

Squib

*Optimality Theory implements complex functions with simple constraints**

Andrew Lamont

University of Massachusetts Amherst

While attested phonological mappings appear to be regular, Optimality Theory is known to implement more complex functions. This squib contributes to the computational characterisation of Optimality Theory by constructing a grammar that implements a non-pushdown function. By using only simple, familiar constraints, the result suggests that a large proportion of optimality-theoretic grammars are more powerful than necessary as models of phonology.

1 Introduction

Non-cyclic rule-based models such as the word-level phonology of Chomsky & Halle (1968) are computationally regular (Johnson 1972, Kaplan & Kay 1994); in other words, their computation does not require unbounded counting. Insofar as these models are empirically adequate, phonology also appears to be regular (see Heinz 2018 for an overview). Optimality Theory (OT; Prince & Smolensky 1993) is known to implement functions whose computations do involve counting (Eisner 1997, Frank & Satta 1998, Karttunen 1998), meaning it overgenerates. However, the total generative capacity of OT with familiar phonological constraints has not yet been characterised. This squib contributes to that characterisation by implementing a function in OT whose computation requires not only counting, but also maintaining multiple unbounded counts. In and of itself, this result is not novel: Lamont (2021) constructs OT grammars that are at least as complex. However, whereas those constructions use markedness constraints whose loci are subsequences, this squib uses substrings constraints. Subsequences are subsets of strings whose elements are not necessarily contiguous at any level of representation. Constraints over subsequences and substrings are the simplest kind of constraints (McNaughton & Papert 1971, Rogers *et al.* 2010,

* E-mail: ALAMONT@LINGUIST.UMASS.EDU.

Heinz 2018), and represent a large proportion of the markedness constraints familiar to phonologists. Together, these results indicate that CON is not the source of OT's generative capacity, and imply that, whenever constraints evaluate candidates by counting an unbounded number of violations, it is possible to construct very complex OT grammars. Further, because this squib uses familiar constraints like AGREE, MAX and IDENT, it may be the case that a large proportion of OT grammars are more powerful than necessary as a model of phonology.

Figure 1 illustrates OT's expressivity on a partial hierarchy of functions. While it successfully models phonological patterns, it also implements non-regular functions, which, by hypothesis, should not be phonological. Note that this does not imply that OT can implement all regular functions. For example, Buccola (2013) proves that certain OT grammars cannot implement counterbleeding-on-focus mappings.

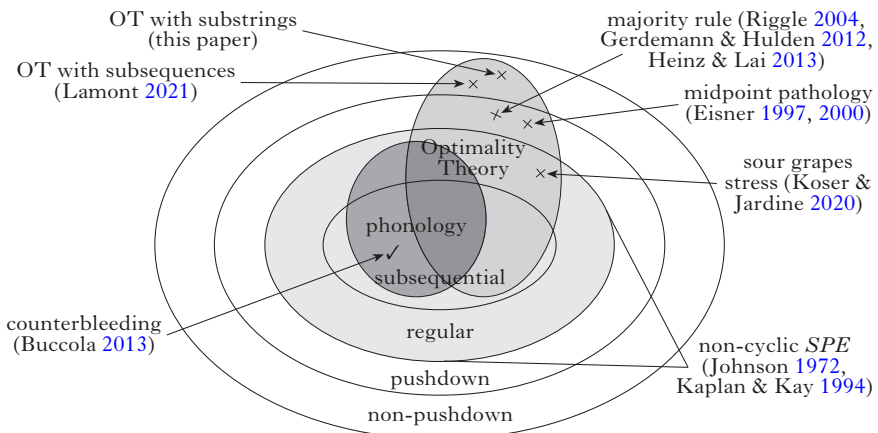


Figure 1

A partial hierarchy of functions illustrating that attested phonological mappings are regular, and that Optimality Theory implements both non-regular and non-pushdown functions.

Regular functions divide inputs into finitely many equivalence classes. For example, 'sour grapes stress' assigns feet only to even-parity words (Koser & Jardine 2020), thus differentiating between inputs with an even or odd number of syllables. Non-regular functions cannot be characterised by a finite number of equivalence classes. For example, consider MAJORITY RULE, an unattested pattern in which disharmonic inputs are mapped onto outputs that preserve the feature value associated with the greater number of input segments (Lombardi 1999, Baković 2000). Majority rule [place] assimilation maps /ttmm/ onto [ttnn] and

/ttmmm/ onto [ppmmm]. Every input $/t^i m^j/$ defines its own equivalence class, and is mapped onto $[t^n]$ when $i > j$, and onto $[p^i m^j]$ when $i < j$. To compute majority rule, it is necessary to compare the number of coronals to the number of labials in the input, which can be modelled by a finite-state machine with unbounded pushdown memory (see Meduna 2000: ch. 9 for an overview). As it reads input strings, such a machine (a PUSHDOWN AUTOMATON) can push a symbol onto the top of its memory or pop the top symbol from its memory. To model majority rule, a machine starts by pushing the first segment it reads to its memory. Then, if it reads a matching segment or if the memory is empty, it pushes that segment, or, if it reads a mismatched segment, it pops the top segment. At every step in the computation, the memory contains n segments of one type, meaning that the machine has seen n more of that segment type so far. At the end of the input, if the memory is empty, then there were an equal number of segment types; otherwise, whichever segment type is stored represents the majority.

Non-pushdown functions, such as the topic of this squib, involve comparing at least three arbitrarily large numbers simultaneously, and cannot be modelled by pushdown automata with only one stack (note that unrestricted pushdown automata with two stacks are equivalent to Turing machines). The intuition is that in order to compare two arbitrarily large numbers, pushdown automata must empty their memory. For example, after reading an input $/t^i m^j/$, the machine's memory will contain $i - j$ coronals or $j - i$ labials, which is sufficient to determine whether i or j is larger. However, pushdown automata cannot identify the majority [place] feature for inputs like $/t^i m^j g^k/$: by comparing the coronals to the labials, the machine forgets how many coronals and labials there were, and has nothing to compare to the dorsals.

This squib implements a non-pushdown function in OT that simultaneously models majority rule [place] and [voice] assimilation. Like the example above, because this function must compare more than two unbounded counts, it is non-pushdown. §2 defines the function of interest and constructs an OT grammar that implements it, proving the main result of the squib. §3 discusses the implications of this result and concludes.

2 A non-pushdown function in Optimality Theory

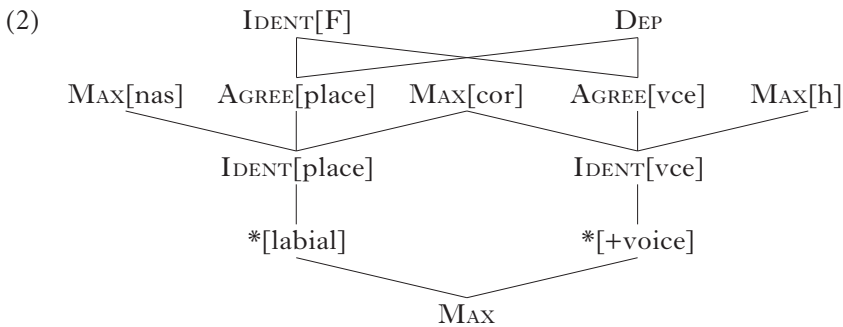
This section proves the main result of the paper by constructing an OT grammar that implements a non-pushdown function. The function of interest is defined in (1) for all inputs $/wh^i x d^j y m^k z/$, where $w, x, y, z \in \{p, b, f\}^*$; the segment inventory is limited to the consonants $\{p, t, b, d, h, f, m, n\}$. In part, the function implements both majority rule [place] and [voice] assimilation; it voices the string of obstruents when there are more voiced obstruents underlyingly ($i < j$) and devoices them otherwise ($i \geq j$), and maps the string of supralaryngeals onto labials when there are

more labials than coronals underlyingly ($j < k$) and onto coronals otherwise ($j \geq k$). The range of this function is not a context-free language: $i < j < k$ for all outputs of the form $[f^i b^j m^k]$ (for a formal proof that this is not a context-free language, see Sipser 2013: 128). Note that the restriction to strings $f^* b^* m^*$ does not increase the complexity of this language. Context-free languages are closed under intersection with regular languages (Bar-Hillel *et al.* 1961), so if the range of the function were context-free, then its intersection with $f^* b^* m^*$ would be too. Because the range of pushdown functions are context-free languages (Evey 1963), it follows that (1) is not a pushdown function.

$$(1) \quad f(\text{wh}^i \text{x}^j \text{d}^k \text{y}^m \text{z}^n) = \begin{cases} f^i b^j m^k, & i < j < k \\ f^i d^j n^k, & i < j \text{ and } j \geq k \\ h^i p^j m^k, & i \geq j \text{ and } j < k \\ h^i t^j n^k, & i \geq j \geq k \end{cases}$$

In addition to implementing majority rule [place] and [voice] assimilation, the grammar deletes underlying /p b f/. Because [p b f] can surface as the result of [place] or [voice] assimilation, this yields a monostratal chain shift, e.g. /t/ → [p] and /p/ → [ε], where [ε] represents the empty string. This is neither beyond the capacity of standard OT (see Baković 2007, Łubowicz 2011 for discussion) nor responsible for the complexity of the function. A variant without deletion can be shown to be non-pushdown, using the pumping lemma for pushdown relations (Gurari 1989: ch. 3). The purpose of the chain shift is to translate function complexity into language complexity, which is often more familiar.

The Hasse diagram in (2) summarises the grammar’s constraint ranking; constraints are defined in the text. The rest of this section presents the grammar. §2.1 illustrates the deletion of underlying /p b f/, §2.2 presents majority rule [place] and [voice] assimilation and §2.3 demonstrates that $i < j < k$ for all output strings $[f^i b^j m^k]$.



2.1 Underlying /p b fi/ delete

Deletion of underlying /p b fi/ is motivated by the constraints *[labial] and *[+voice], which penalise labial supralaryngeals ([p b m]) and voiced obstruents ([b d fi]) respectively. Deletion violates MAX, which requires segments in the input to have correspondents in the output. Specific MAX constraints prevent other segments from deleting: MAX[nas] and MAX[cor] respectively require input nasals (/m n/) and input coronals (/t d n/) to have output correspondents. In isolation, IDENT[place] and IDENT[vce] respectively prevent underlying labials from surfacing as coronals and underlying voiced segments from devoicing. IDENT[place] penalises pairs of correspondent supralaryngeals ({p, t, b, d, m, n}) with different specifications of [place]. IDENT[vce] penalises pairs of correspondent obstruents ({p, t, b, d, h, fi}) with different specifications of [voice].

The tableaux in (3) illustrate the deletion of underlying /p, b/, but not /m/. For each input, a set of winner ~ loser pairs and their evaluation by the relevant constraints are shown. The faithful realisation of /p/ and /b/ violates *[labial], and is dispreferred to deletion (3a.i, b.i). Mapping /p/ onto [t] and /b/ onto [d] also satisfies *[labial], but fatally violates IDENT[place] (3a.ii, b.ii). Deleting underlying /b/ is also preferred by *[+voice] (3b.i, ii). Devoicing would also satisfy *[+voice], but fatally violates IDENT[vce] (3b.iii, iv). Underlying /m/ surfaces faithfully (3c), because deleting underlying nasals fatally violates MAX[nas] (3c.i) and mapping /m/ onto [n] fatally violates IDENT[place] (3c.ii).

(3) a.	/p/	MAX[nas]	IDENT[place]	*[labial]	IDENT[vce]	*[+voice]	MAX
i.	$\epsilon \sim p$			W			L
ii.	$\epsilon \sim t$		W				L
b.	/b/						
i.	$\epsilon \sim b$			W		W	L
ii.	$\epsilon \sim d$		W			W	L
iii.	$\epsilon \sim p$			W	W		L
iv.	$\epsilon \sim t$		W		W		L
c.	/m/						
i.	$m \sim \epsilon$	W		L			W
ii.	$m \sim n$		W	L			

Along similar lines, *[+voice] motivates the deletion of underlying /fi/, but not /d/ (4). Devoicing either segment fatally violates IDENT[vce] (4a.ii, b.ii), and MAX[cor] prevents underlying /d/ from deleting (4b.i). Because *[+voice] does not penalise nasals, another option would be to map /d/ onto [n]. Though not shown in the tableau, this is ruled out by

the cover constraint IDENT[F], which penalises pairs of correspondent segments with different specifications of any feature other than [place] or [voice].

(4) a.

/fɪ/	MAX[cor]	IDENT[vce]	*[+voice]	MAX
i. $\epsilon \sim \text{fi}$			W	L
ii. $\epsilon \sim \text{h}$		W		L

b.

/d/				
i. $\text{d} \sim \epsilon$	W		L	W
ii. $\text{d} \sim \text{t}$		W	L	

Because underlying /p b fi/ delete, no segments in the output correspond to /p b fi/ segments in the input. The set of possible input correspondents is tightened further by IDENT[F] and DEP, which require segments in the output to have correspondents in the input. Ranking DEP above the grammar's markedness constraints means that epenthesis cannot be optimal. Thus, because no segments in the output are epenthetic, they must all correspond to underlying /t d h m n/. IDENT[F] also dominates all markedness constraints, preventing any featural change except to [place] or [voice]. Together, these limitations on input–output correspondents mean that [p t b d] in the output must correspond to /t d/ in the input, [h fi] in the output must correspond to /h/ in the input and [m n] in the output must correspond to /m n/ in the input.

2.2 Majority rule controls [place] and [voice] assimilation

While *[labial] and *[+voice] motivate the deletion of underlying /p b fi/, they do not prevent [p b fi] from surfacing via [place] and [voice] assimilation (hence the chain shift). Assimilation is motivated by the constraints AGREE[place] and AGREE[vce], which respectively penalise adjacent supralaryngeals ([p t b d m n]) with different specifications of [place] and adjacent obstruents ([p t b d h fi]) with different specifications of [voice].

The tableaux in (5) illustrate [place] assimilation targeting underlying /dⁱm^j/ strings of various lengths. The faithful candidates violate AGREE[place], and are dispreferred to candidates that map /dⁱ/ onto [bⁱ] (5a.i) or /m^j/ onto [n^j] (5b.i, c.i). Deleting either underlying /m^j/ strings (5a.ii, b.ii, c.ii) or /dⁱ/ strings (5a.iii, b.iii, c.iii) also satisfies AGREE[place], but fatally violates MAX[nas] or MAX[cor]. Likewise, inserting a laryngeal ([dⁱh^mj]; not shown), satisfies AGREE[place], but is ruled out by DEP. Whether it is optimal to map /dⁱ/ onto [bⁱ] or /m^j/ onto [n^j] is determined by the relative magnitudes of *i* and *j*. When there are fewer coronals than labials (*i* < *j*), /dⁱ/ is mapped onto [bⁱ] (5a.iv), because IDENT[place] prefers making as few changes as possible. When there are fewer labials than

coronals ($i > j$), $/m^j/$ is mapped onto $[n^j]$ (5c.iv). With equal numbers of labials and coronals ($i = j$), IDENT[place] does not prefer either mapping, and the decision falls to *[labial], which prefers mapping $/m^j/$ onto $[n^j]$ (5b.iv). As in previous tableaux, devoicing the underlying $/d^i/$ satisfies *[+voice], but fatally violates IDENT[vce] (5a.v, b.v, c.v).

(5) a.

$/d^i m^j/$ $i < j$	AGREE [place]	MAX [nas]	MAX [cor]	IDENT [place]	*[labial]:IDENT [vce]	*[+voice]	MAX
i. $b^i m^j \sim d^i m^j$	W			L	L		
ii. $b^i m^j \sim d^i$		W		L	L		W
iii. $b^i m^j \sim m^j$			W	L	L	L	W
iv. $b^i m^j \sim d^i n^j$				W	L		
v. $b^i m^j \sim p^i m^j$					W	L	

b.

$/d^i m^j/$ $i = j$							
i. $d^i n^j \sim d^i m^j$	W			L	W		
ii. $d^i n^j \sim d^i$		W		L			W
iii. $d^i n^j \sim m^j$			W	L	W	L	W
iv. $d^i n^j \sim b^i m^j$					W		
v. $d^i n^j \sim t^i n^j$					W	L	

c.

$/d^i m^j/$ $i > j$							
i. $d^i n^j \sim d^i m^j$	W			L	W		
ii. $d^i n^j \sim d^i$		W		L			W
iii. $d^i n^j \sim m^j$			W	L	W	L	W
iv. $d^i n^j \sim b^i m^j$				W	W		
v. $d^i n^j \sim t^i n^j$					W	L	

Along similar lines, AGREE[vce] motivates majority rule assimilation within strings of obstruents ($\{p, t, b, d, h, \text{fi}\}$). The tableaux in (6) illustrate this with $/h^i d^j/$ strings of various lengths. As above, MAX[cor] prevents underlying $/d^j/$ from deleting (6a.ii, b.ii, c.ii), and MAX[h], which requires $/h/$ in the input to have a correspondent in the output, prevents underlying $/h^i/$ from deleting (6a.iii, b.iii, c.iii). DEP prevents the insertion of a nasal ($[h^i n d^j]$; not shown). When there are fewer underlying voiceless laryngeals ($i < j$), they are voiced (6a), and when there are at least as many voiceless laryngeals as voiced coronals ($i \geq j$), the coronals devoice (6b, c), with *[+voice] breaking any ties in favour of devoicing (6b.iv). Mapping $/d^j/$ onto $[n^j]$ (not shown) would satisfy AGREE[vce] and *[+voice], but is ruled out by IDENT[F].

(6) a.

$/h^i d^j/$ $i < j$	AGREE [vce]	MAX [cor]	MAX [h]	IDENT [vce]	*[+voice]	MAX
i. $f^i d^j \sim h^i d^j$	W			L	L	
ii. $f^i d^j \sim h^i$		W		L	L	W
iii. $f^i d^j \sim d^j$			W	L	L	W
iv. $f^i d^j \sim h^i t^j$				W	L	

b.

$/h^i d^j/$ $i = j$						
i. $h^i t^j \sim h^i d^j$	W			L	W	
ii. $h^i t^j \sim h^i$		W		L		W
iii. $h^i t^j \sim d^j$			W	L	W	W
iv. $h^i t^j \sim f^i d^j$					W	

c.

$/h^i d^j/$ $i > j$						
i. $h^i t^j \sim h^i d^j$	W			L	W	
ii. $h^i t^j \sim h^i$		W		L		W
iii. $h^i t^j \sim d^j$			W	L	W	W
iv. $h^i t^j \sim f^i d^j$				W	W	

As argued in the previous section, surface [p b] must derive from underlying /t d/, and surface [f] must derive from underlying /h/. The only mechanism that motivates these mappings is majority rule [place] or [voice] assimilation. Thus, for [p b f] to surface, their input correspondents must have had the minority specification of [place] or [voice]. It follows that, for example, $i < j$ for all strings $[b^i m^j]$ in the output, and $k < l$ for all strings $[f^k d^l]$ in the output. In isolation, these string-sets are context-free, but their intersection is not (Scheinberg 1960). In other words, string-sets produced by this grammar are not context-free languages.

2.3 Simultaneous majority rule is non-pushdown

To complete the argument, consider output strings of the form $[f^i b^j m^k]$. According to the discussion in the previous sections, these strings must derive from underlying $/... h^i ... \{t, d\}^j ... \{m, n\}^k .../$, which may also contain arbitrarily many /p b f/ segments. It must be true that $i < j < k$. Let h, t, d, m, n denote the number of /h t d m n/ in the input, respectively. Observe that $h = i, t + d = j$, and $m + n = k$. Majority rule [voice] assimilation produces voiced obstruents, and so the input must contain more voiced obstruents than voiceless: $h + t < d$. Majority rule [place] assimilation produced labial supralaryngeals, and so the input must have contained more labials than coronals: $t + d + n < m$. These inequalities combine to $h \leq h + t < d \leq t + d \leq t + d + n < m \leq m + n$, which implies $h < t + d < m + n$, or, equivalently, $i < j < k$. Therefore, because the range of the function implemented by this grammar is not a context-free language, the function is non-pushdown.

The tableau in (7) illustrates $/h^i d^j m^k/$ mapping onto $[fi^i b^j m^k]$, with $i < j < k$. The faithful candidate (7a) violates both AGREE constraints, and is dispreferred to the candidate that maps $/h^i/$ onto $[fi^i]$ and $/d^j/$ onto $[b^j]$ (7f). Deleting both $/h^i/$ and $/m^k/$ also satisfies the AGREE constraints (7b), as does deleting $/d^j/$ (7c), but fatally violates the input-specific MAX constraints. Because there are more labial supralaryngeals than coronal underlyingly, IDENT[place] prefers mapping $/d^j/$ onto $[b^j]$ or $[p^j]$ (7f, g) to mapping $/m^k/$ onto $[n^k]$ (7d, e). Likewise, because there are more voiced obstruents than voiceless underlyingly, IDENT[vce] prefers mapping $/h^i/$ onto $[fi^i]$ (7d, f) to mapping $/d^j/$ onto $[t^j]$ or $[p^j]$ (7e, g). Per (1), candidates (7d, e, g) are optima for other relative values of i, j, k : the supralaryngeals surface as coronals when $j \geq k$ (7d, e), and the obstruents surface as voiceless when $i \geq j$ (7e, g). Though not shown in (7), IDENT[F] and DEP prevent other mappings from being optimal.

(7)

$/h^i d^j m^k/$ $i < j < k$	AGREE [pl]	AGREE [vce]	MAX [nas]	MAX [cor]	MAX [h]	ID [pl]	*[lab] [vce]	ID [vce]	*[+vce]	MAX
a. $h^i d^j m^k$	W1	W1				L	Lk	L	Lj	
b. d^j			Wk		Wi	L	L	L	Lj	Wi+k
c. $h^i m^k$				Wj		L	Lk	L	L	Wj
d. $fi^i d^j m^k$						Wk	L	i	i+j	
e. $h^i t^j n^k$						Wk	L	Wj	L	
f. $fi^i b^j m^k$						j	j+k	i	i+j	
g. $h^i p^j m^k$						j	j+k	Wj	L	

3 Discussion

The OT grammar constructed in this squib implements a function that deletes underlying $/p b fi/$ and subjects supralaryngeals ($\{p, t, b, d, m, n\}$) to majority rule [place] assimilation and obstruents ($\{p, t, b, d, h, fi\}$) to majority rule [voice] assimilation. This function was shown to be non-pushdown by demonstrating that its range is not a context-free language. Assuming that phonology is regular, OT significantly overgenerates, even with simple constraints. Further, it is possible to generalise this construction to model majority rule over arbitrarily many features, implying that OT is beyond the generative capacity of tree-adjointing grammars (Joshi 1985). While this construction assumes a finite feature inventory, it may be possible to accommodate an unbounded feature set and thereby model functions whose ranges are even more complex (Radzinski 1991, Michaelis & Kracht 1997).

Computational analysis of OT has often focused on constraints on subsequences, because their loci of violation grow quadratically in the length of candidates (Eisner 1997, 2000, Bíró 2003, Heinz 2005, Lamont 2021),

which in and of itself is not a regular relation. However, as this squib has demonstrated, quadratic growth is not necessary for OT to implement complex functions, as substrings grow linearly in the length of candidates. Notably, if the IDENT constraints employed in the construction could not distinguish between arbitrarily large violation sets, the grammar would be much less expressive, mapping all inputs onto a set of least marked outputs similar to Chomsky's (2015: 350) [ba] problem. This reaffirms the result that unbounded counting is the source of OT's complexity (Eisner 1997, Frank & Satta 1998, Karttunen 1998): by freely combining constraints that count unboundedly, OT grammars can be very complex. Further, as Hao (2019) demonstrates, switching from parallel to serial evaluation does not address the problem; constraints on substrings also motivate non-regular mappings in Harmonic Serialism.

Studying the generative capacity of OT characterises it as a computational framework, while abstracting away from particular theories of CON. Comparison to the expressivity of attested phenomena is informative, because it provides a coarse view of under- and overgeneration. As discussed in the introduction, non-cyclic rule-based models are known to be regular, and are consistent with the hypothesis that phonology is regular. Because OT implements non-regular functions, it appears to overgenerate, from our current empirical perspective. This approach is distinct from evaluating other instances of overgeneration, such as the too-many-repairs problem (Wilson 2000, Blumenfeld 2006, Steriade 2009). In particular, the grammar in this squib was constructed specifically to implement a complex function, and not to propose that, for example, MAX[h] is an empirically desirable or undesirable constraint. Banning particular constraints may accidentally rein in OT's expressivity, but such an approach fails to address the problem of unbounded violation counting, which is a core component of OT. It would therefore be a quixotic exercise to manipulate CON without solving this underlying issue. Results like this squib's contribute to our understanding of OT, and point to alternative approaches to optimisation, such as bounded evaluation (Frank & Satta 1998) or directional evaluation (Eisner 2000, 2002), whose computational restrictiveness explains why majority rule and other complex mappings should be impossible.

REFERENCES

- Baković, Eric (2000). *Harmony, dominance and control*. PhD dissertation, Rutgers University.
- Baković, Eric (2007). A revised typology of opaque generalisations. *Phonology* 24. 217–259.
- Bar-Hillel, Y., M. Perles & E. Shamir (1961). On formal properties of simple phrase structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung* 14. 143–172.
- Biró, Tamás (2003). Quadratic alignment constraints and finite state Optimality Theory. In *Proceedings of the Workshop on Finite-State Methods in Natural Language Processing (FSMNLP), 10th Conference of the European Chapter of the Association for Computational Linguistics*. 119–126.

- Blumenfeld, Lev (2006). *Constraints on phonological interactions*. PhD dissertation, Stanford University.
- Buccola, Brian (2013). On the expressivity of Optimality Theory versus ordered rewrite rules. In Glyn Morrill & Mark-Jan Nederhof (eds.) *Formal grammar: Proceedings of the 17th and 18th International Conferences, FG 2012/2013*. Berlin & Heidelberg: Springer. 142–158.
- Chomsky, Noam (2015). *The minimalist program (20th anniversary edition)*. Cambridge, MA: MIT Press.
- Chomsky, Noam & Morris Halle (1968). *The sound pattern of English*. New York: Harper & Row.
- Eisner, Jason (1997). What constraints should OT allow? Handout from paper presented at the 71st Annual Meeting of the Linguistic Society of America, Chicago. Available as ROA-204 from the Rutgers Optimality Archive.
- Eisner, Jason (2000). Directional constraint evaluation in Optimality Theory. In *COLING 2000: proceedings of the 18th Conference on Computational Linguistics, Saarbrücken*. Vol. 1. San Francisco: Morgan Kaufmann. 257–263.
- Eisner, Jason (2002). Comprehension and compilation in optimality theory. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 56–63.
- Evey, R. James (1963). *The theory and applications of pushdown store machines*. PhD dissertation, Harvard University.
- Frank, Robert & Giorgio Satta (1998). Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24. 307–315.
- Gerdemann, Dale & Mans Hulden (2012). Practical finite state Optimality Theory. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*. Association for Computational Linguistics. 10–19.
- Gurari, Eitan M. (1989). *An introduction to the theory of computation*. Rockville, MD: Computer Science Press.
- Hao, Yiding (2019). Finite-state Optimality Theory: non-rationality of Harmonic Serialism. *Journal of Language Modelling* 7:2. 49–99.
- Heinz, Jeffrey (2005). Reconsidering linearity: evidence from CV metathesis. *WCCFL* 24. 200–208.
- Heinz, Jeffrey (2018). The computational nature of phonological generalizations. In Larry M. Hyman & Frans Plank (eds.) *Phonological typology*. Berlin & Boston: De Gruyter Mouton. 126–195.
- Heinz, Jeffrey & Regine Lai (2013). Vowel harmony and subsequentality. In András Kornai & Marco Kuhlmann (eds.) *Proceedings of the 13th Meeting on the Mathematics of Language*. Sofia: Association for Computational Linguistics. 52–63.
- Johnson, C. Douglas (1972). *Formal aspects of phonological description*. The Hague & Paris: Mouton.
- Joshi, Aravind K. (1985). Tree adjoining grammars: how much context-sensitivity is required to provide reasonable structural descriptions? In David R. Dowty, Lauri Karttunen & Arnold M. Zwicky (eds.) *Natural language parsing: psychological, computational, and theoretical perspective*. Cambridge: Cambridge University Press. 206–250.
- Kaplan, Ronald M. & Martin Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* 20. 331–378.
- Karttunen, Lauri (1998). The proper treatment of optimality in computational phonology. In Lauri Karttunen & Kemal Oflazer (eds.) *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*. Ankara: Bilkent University. 1–12.
- Koser, Nate & Adam Jardine (2020). The complexity of optimizing over strictly local constraints. *University of Pennsylvania Working Papers in Linguistics* 26:1. 125–134.

- Lamont, Andrew (2021). Optimizing over subsequences generates context-sensitive languages. *Transactions of the Association for Computational Linguistics* **9**. 528–537.
- Lombardi, Linda (1999). Positional faithfulness and voicing assimilation in Optimality Theory. *NLLT* **17**. 267–302.
- Lubowicz, Anna (2011). Chain shifts. In Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume & Keren Rice (eds.) *The Blackwell companion to phonology*. Malden, MA: Wiley-Blackwell. 1691–1716.
- McNaughton, Robert & Seymour A. Papert (1971). *Counter-free automata*. Cambridge, MA: MIT Press.
- Meduna, Alexander (2000). *Automata and languages: theory and applications*. London: Springer.
- Michaelis, Jens & Marcus Kracht (1997). Semilinearity as a syntactic invariant. In Christian Retoré (ed.) *Logical aspects of computational linguistics*. Berlin: Springer. 329–345.
- Prince, Alan & Paul Smolensky (1993). *Optimality Theory: constraint interaction in generative grammar*. Ms, Rutgers University & University of Colorado, Boulder. Published 2004, Malden, MA & Oxford: Blackwell.
- Radzinski, Daniel (1991). Chinese number-names, tree adjoining languages, and mild context-sensitivity. *Computational Linguistics* **17**. 277–300.
- Riggle, Jason (2004). *Generation, recognition, and learning in finite-state Optimality Theory*. PhD dissertation, University of California, Los Angeles.
- Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome & Sean Wibel (2010). On languages piecewise testable in the strict sense. In Christian Ebert, Gerhard Jäger & Jens Michaelis (eds.) *The mathematics of language*. Berlin & Heidelberg: Springer. 255–265.
- Scheinberg, Stephen (1960). Note on the Boolean properties of context free languages. *Information and Control* **3**. 372–375.
- Sipser, Michael (2013). *Introduction to the theory of computation*. 3rd edn. Boston, MA: Cengage Learning.
- Steriade, Donca (2009). The phonology of perceptibility effects: the P-map and its consequences for constraint organization. In Kristin Hanson & Sharon Inkelas (eds.) *The nature of the word: studies in honor of Paul Kiparsky*. Cambridge, MA: MIT Press. 151–179.
- Wilson, Colin (2000). *Targeted constraints: an approach to contextual neutralization in Optimality Theory*. PhD dissertation, Johns Hopkins University.