On the problem of Infinity and Gen-Eval in Optimality Theory

Paul Hagstrom, MIT
November 1993

In this squib, I present an alternative view of the Gen and Eval mechanisms that lie at the heart of Optimality Theory ("OT") (see Prince and Smolensky 1993, McCarthy and Prince 1993). As originally conceived, Gen is a pattern generator which creates an infinite set of possible phonological realizations of a lexical input string ("candidates"). Eval then evaluates the entire (infinite) set to determine which candidate is optimal with respect to a finite set of phonological constraints. The claim of OT is that these constraints are given in Universal Grammar ("UG") and that only the ordering of the constraints differs from language to language.

The problem of infinity

Any in-depth consideration of the Gen-Eval mechanism immediately brings up a question as to how it is computationally possible to conduct an evaluation over an infinite set in finite time. Given our current concepts of algorithmic computation, this appears to be a serious problem. Prince and Smolensky (1993) answer this and similar criticism by dismissing it as an imposition of "arbitrary meta-constraints (e.g. 'computational plausibility') which could conflict with the well-defined basic goals of the enterprise."[1] Still, there are grounds for uneasiness with a framework which requires its mechanisms to function in ways so far beyond any current notions of computability. Here, I suggest that by looking at OT from a different perspective may allow us to discard the "infinity problem," while still retaining the core principles behind Violability, Ranking, Inclusiveness, and Parallelism.[2]

A possible solution involving weighted constructors instead of constraints

Setting as our goal an elimination of the "infinity problem," there are a couple of simple patches we could make to the framework. One such patch would attribute an internal constraint to Gen which limits the amount of structure it is allowed to add to the input form, thus solving the infinity problem. Another patch might suggest that Gen "short circuits" the evaluation process by generating (and evaluating) first the candidates which are most likely to succeed. Neither of these approaches is appealing, though; the first solution seems to be simply hiding the problem behind an explanatorily vacuous limit, while the second is doomed from the outset if we assume that the constraints may occur in an arbitrary order.

Instead, I suggest looking at the Gen-Eval mechanism in a "backwards" way: rather than viewing the constraints as ordered filters in Eval, we might instead be able to use them as weighted constructors of the optimal candidate. Doing this eliminates the infinity problem, since there are a finite number of constructors (each analogous to an OT constraint), and only one candidate (the most optimal) is actually constructed.

To show that this "backwards" approach to OT has some merit, we will need to look at how it functions under a specific set of assumptions about representational structure in the phonological component of language.

---

[1] Prince and Smolensky (1993), §10.1.2.
[2] From McCarthy and Prince (1993), §2, in a short overview of Optimality Theory.

Accordingly, this view of OT is presented as a suggestion of a possible line of research; too many assumptions are required for this to be a proof of equivalence to the standard OT framework.

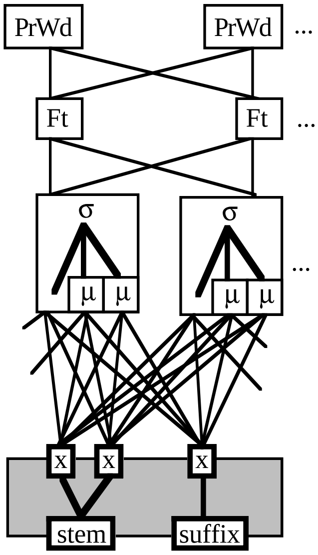

Fig. 1. Interconnected prosodic template

At the center of the constructor-based approach to OT is an interconnected template for prosodic structure, similar to that shown in figure 1. The levels in the shaded box are presumed to come directly from the lexicon, and the connections therein are assumed to be previously fixed and unalterable by the phonological constructors. The symbol "x" represents a phonological segment, and the which is connected to the onset and nucleus of each syllable is epenthetic material, which is allowed to participate in the connection weighting process along with the lexical segments. The constructors operate on this structure by strengthening or weakening connections, and the "ordering" (priority) of the constraints is achieved by giving the different constructors different strengths or weights. All connections in the network will initially be equally weighted; let us say they begin at an activation level of zero (a zero "bias"). The constructors will then reweight the connections simultaneously.

As an initial aside, I want to stress that by choosing this weighted interconnection model, I am not making any claims about its implementation; specifically, I am not claiming that this system is more elegant by any "neural plausibility" standards. This is not presented as a "connectionist implementation of OT," but is intended as an alternative view of the Gen-Eval mechanism that eliminates the need for evaluation over an infinite set of candidates.

In figure 2 below, I have listed some constructor analogs to selected OT constraints. In each case, the number N is to be taken as the "weight" of the individual constructor, which correlates with the ranking of the constraints in the standard OT framework. In the particular case of the HNUC constructor, I also assume a numerical sonority hierarchy that is lexically specified. In other words, the lexicon is assumed to have a "sonority value" for each segment, and that value is used by HNUC for weighting. I will further assume zero sonority to be the "syllable nucleus cutoff point," such that only segments with a positive "sonority value" will be permissible syllable nuclei in the language.

ONSET (Encourage syllable onsets):
+N to all connections to syllable onset.

HNUC (Encourage sonorous nuclei for syllables):
+S to all connections between segments and nuclei, where S is the lexically specified sonority value of the segment.

FILL (Discourage epenthesis):
-N to all connections to epenthetic material.

PARSE (Encourage full parsing of segments):
+N to all connections from segments to syllables.

NO-CODA (Discourage syllable codas):
-N to all connections to syllable codas.

CODA-COND (Encourage homorganic coda-onsets):
+N to all segment-syllable connections for homorganic segment pairs having one linked to the coda of a syllable and the other linked to the onset of the following syllable

Figure 2. Constructor analogs to standard Optimality Theory constraints

Once the constructors have had their chance to reweight the tree, a finalization process will select the most highly activated connections and those connections compatible with them. If we maintain commonly assumed tree well-formedness requirements like “no crossing branches” and “no doubly linked items” then by finalizing the most highly activated connections, some connections will become disqualified regardless of their activation level on ill-formedness grounds. Finalization is therefore assumed to occur sequentially from most highly activated connections to least activated connections, and to follow principles such as those listed in figure 3 below.

Tree well-formedness:

- When a link between a syllable and a segment is finalized, all other links to that segment are disqualified and all other links from the syllable node are disqualified.

Lexical material must be continuous (no crossing branches):

- When a link between a syllable and a segment is finalized, all links that would link preceding segments to following syllable nodes or link preceding syllable nodes to following segments are disqualified.

All syllables must have a nucleus:

- If no connections remain with the nucleus of a syllable, the entire syllable is disqualified.
- The last connection with a syllable nucleus cannot be disqualified if a connection with a lexical segment has previously been finalized within the syllable.

Prefer most compact representation:

- In the event of a tie for the most highly activated connection, finalize the leftmost connection.

Figure 3. Some principles of finalization for segment-syllable connections.

An example

As a very simple example of the approach, we can look at how a lexical input form like /CVC/ can be syllabified under two different configurations: HNUC>>NOCODA>>FILL and HNUC>>FILL>>NOCODA. As adapted from a discussion in McCarthy and Prince (1993), these should yield CV.C and .CVC., respectively. In figures 4 and 5 below, the process is exhibited. Note that the connections between the syllables and the segments are separated from one another for readability; the first four columns represent the connections between the onset of the first syllable and each of the input segments, the last of which is the structure-internal epenthetic segment. Notice also that we are only looking at the relevant subset of the total connections, namely those between the syllable level and the lexical segment level shown in figure 1. Immediately below the connection labels in the figures below is the “final weight” of the connection, using the first letter of the constraint as the label (“+h” means “strengthened by HNUC,” while “-f” means “weakened by FILL.”). The last few lines in the diagrams show steps in the finalization process. There, “X” represents a newly finalized connection, “*” represents a newly disqualified connection (usually by virtue of the recent finalization), and “x” and “.” represent finalizations and disqualifications that happened in a previous step. Figure 4 shows the situation in which HNUC>>NOCODA>>FILL.

					σ											σ					
	onset				nucleus				coda		onset				nucleus				coda		
C	V	C	_	C	V	C	_	C	V	C	C	V	C	_	C	V	C	_	C	V	C
			-f	-h	+h	-h	-f	-n	-n	-n				-f	-h	+h	-h	-f	-n	-n	-n
	*	*		*	X	*	*	*	*		*	*			*	*	*		*	*	
X	.	.	*	.	x	.	.	.	.		.	.			.	.	.		.	.	
x	.	.	.	.	x	.	.	.	.	*	.	.	X	*	.	.	.		.	.	*
x	.	.	.	.	x	.	.	.	.	.	.	.	x	.	.	.	.	X	.	.	.

Figure 4. Finalization process for HNUC>>NOCODA>>FILL.

Reading the finalization process in figure 4, we see that the positive HNUC connection is made first and all incompatible connections are disqualified, which includes all other connections to the nucleus of the first syllable, all other connections to the lexical segment /V/, and all connections which would result in discontinuous lexical material. In the second line, there is a tie for most highly activated connection, so the leftmost one is chosen, finalizing the connection between the first lexical /C/ and the onset of the first syllable. The third line then finalizes the connection between the last lexical /C/ and the second syllable, and the fourth line finalizes the epenthetic nucleus of the second syllable. The result is .CV.C , as desired.

The second diagram, figure 5, shows the case where HNUC>>FILL>>NOCODA.

					σ											σ					
	onset				nucleus				coda		onset				nucleus				coda		
C	V	C	_	C	V	C	_	C	V	C	C	V	C	_	C	V	C	_	C	V	C
			-f	-h	+h	-h	-f	-n	-n	-n				-f	-h	+h	-h	-f	-n	-n	-n
	*	*		*	X	*	*	*	*		*	*			*	*	*		*	*	
X	.	.	*	.	x	.	.	.	.		.	.			.	.	.		.	.	
x	.	.	.	.	x	.	.	.	.		.	.		*	.	.	.	*	.	.	
x	.	.	.	.	x	.	.	.	.	X											

Figure 5. Finalization process for HNUC>>FILL>>NOCODA.

The finalization process in figure 5 differs from that in figure 4 only after the second line, where since FILL-weakened connections are more strongly dispreferred, the second syllable's onset and nucleus connections with epenthetic materials are disqualified. By the third of our finalization principles in figure 3, this has the effect of eliminating the second syllable from consideration entirely, since syllables cannot exist without nuclei and all possible nuclei for the second syllable have been eliminated. The next line shows the finalization of the only remaining connection, yielding the desired result: .CVC.

Comments and speculations

Clearly, there are many obstacles to consider and overcome which remain for those who would pursue this constructor-based approach to OT. For one thing, many constraints currently used in OT still need to have constructor analogs formulated. Examples include FTBIN, which enforces minimally bimoraic feet, and the constraints in the ALIGN family (ALIGN-L, ALIGN, SFX-PRWD), which align prosodic constituents. While OT constraints like PARSE and ONSET had obvious analogs, these remaining constraints have a number of possible instantiations in a constructor framework. For example, to formulate FTBIN, we need to consider how best to enforce bimoraicity of prosodic feet. Perhaps we would want to assume two upward connections from each syllable, one for each mora and somehow ensuring, perhaps through finalization principles, that these connections are severed if the associated morae are unfilled. Then, we might propose two downward connections from each foot which FTBIN would encourage to be connected. For other constraints like ALIGN(Stem,L,PrWd,L), we need a more complete theory of the initial interconnected prosodic template in order to determine which connections between the left edge of a PrWd and the left edge of a stem are available for strengthening. Questions arise like "Must a PrWd be connected to a foot, or can it be connected directly to a syllable?" These questions need to be answered by research which is far beyond the scope of the present squib.

It is also important to consider whether the spirit of OT has been retained. Is this simply another way to view OT, as claimed, or is it really a separate theory altogether? I answer this by considering the four principles of OT: Violability, Ranking, Inclusiveness, and Parallelism. Violability and Ranking seem unaffected by the shift in viewpoints: not all constructors will successfully promote a structure which is optimal from their point of view, a direct by-product of weighting, or ranking. We can still think of constructors as given by UG and ranked on a language-particular basis. Inclusiveness entails that only general considerations of structural well-formedness affect the analyses, without recourse to rules or repair strategies. Both constructor and constraint frameworks meet the Inclusiveness requirement. The final principle, Parallelism, envisions the entire evaluation process as a simultaneous computation, which is emphatically not a derivation. Despite that fact that distinct stages are assumed to exist in the finalization process, it is probable that the process of "choosing the most highly preferred connections to the exclusion of incompatible connections" isn't really a "derivation" of the sort that OT wishes to avoid. It seems reasonable to conclude that, despite significant surface differences, both the constructor and constraint frameworks are "in the spirit" of OT.

With a more developed constructor-based approach to OT, we can avoid the "infinity problem" which troubles the constraint-based framework. Further, if we require that the only action allowed to constructors is the strengthening or weakening of prosodic connections, we can constrain the realm of possible OT constructors.

With a possible answer to the infinity problem described, I will now reverse and encourage continuation of *constraints-based* OT research. It is probable that attempting to work directly with connection networks in a constructor-based OT would most likely be too "reductionistic," and would hamper serious OT research. Better would be to retain in the back of our minds this idea of constructors, attempting a move to a constructor-based OT only after the constraint-based framework has a strong foothold. At that time, results from constraints-based OT will be invaluable for determining the appropriate structure for a feasible constructor-based theory, where present knowledge may be too incomplete to avoid a theoretic garden path. Thus, the view of OT proposed here should be taken mainly as comfort and as a possible research program to pursue at some time in the non-immediate future.

References

Prince and Smolensky (1993), Optimality Theory, unpublished ms.
McCarthy and Prince (1993), Prosodic Morphology I, unpublished ms.