Operations are Intrinsically Ordered in Harmonic Serialism

Nazarré Merchant
Eckerd College, St. Petersburg, FL

Harmonic Serialism (HS) (P&S 1993/2004) differs definitionally in two ways from Optimality Theory (OT) (P&S 1993/2004): the output of Gen_{HS} is defined by a set of operations, and the production of a final output is serial, in that the winner of a competition becomes the input to the next, proceeding until the output is equal to the input (see McCarthy 2000, McCarthy 2008, Pater 2010, amongst others, for the consequences of this definitional difference). As is well understood, in OT, given a theory of Gen_{OT} and a set of constraints, a total order on the constraints will define a language. In HS a total order on the constraints will also define a language, but for a given grammatical theory of HS, *a priori*, the *operations* are not endowed with order as part of their definition; every operation that is applicable to a given input is applied with no operation privileged over any other, and then these operation-produced candidates compete for optimality in the usual way. In this talk I show (1) that operations can be categorized and decomposed into *operation vectors* ('opvecs'), integer-valued vectors over the constraints and (2) that each language imposes a partial order on these operation vectors in such a way that if a set of operations produces competing candidates the operation having the highest ranked operation vector, according to the partial order, will have produced the optimal candidate. Knowing the partial order for a language provides complete information about which operation produces the winning candidate in any given situation.

The partial ordering that arises for a given language relies on the construction of operation vectors for each operation. The opvecs are derived from the derivationally produced candidate sets by fixing a single candidate in each set as the base (the fully faithful candidate serves admirably in this role) and then subtracting the base's violation profile from each competitor's violation profile. Each so-produced vector is an opvec for the operation that constructed that competitor. Using these opvecs, each candidate in the candidate set may then be generated as base+opvec, rather than by consulting Gen directly.

This calculation is easily seen in a simple example representing basic syllable theory. Suppose the HS system has inputs of C's and V's; four constraints, Onset, NoCoda, Max, and Dep; and two operations, epenthesis and deletion. Further suppose that we are investigating the language L_{Epen} produced from the total order Onset >> NoCoda >> Max >> Dep. Then given the input /CVC/ we have the candidate set given in Tableau 1 along with the opvecs for epenthesis and deletion in Tableau 2.

Tableau 1.

	/CVC/	**Onset**	**NoCoda**	**Max**	**Dep**	**Operation**
(1)	.CVC.	0	1	0	0	Faithful
(2)	.CV.	0	0	1	0	Deletion
(3)	.CV.CV.	0	0	0	1	Epenthesis

Tableau 2.

		Onset	NoCoda	Max	Dep
(1) - (1)	Faith vector	0	0	0	0
(2) - (1)	Del vector	0	-1	1	0
(3) - (1)	Epen vector	0	-1	0	1

The language then induces a partial order on these opvecs, defined as follows. A vector α is greater than β with respect to language L if α and β are produced from competing candidates *a* and *b*, respectively, and the language selects *a* over *b*. In the case above the language L_{Epen} orders the epenthesis vector above the deletion vector since candidate (3) is optimal. Because this relation can be shown to be a partial order, we can conclude that in L_{Epen}, when faced with these same op vectors in other derivations with other inputs, will select the epenthetic candidate, whatever its form, over the deletional.

Having this partial order on hand concisely represents and determines the operation choices a language will make in any given situation. Even though the order on the opvecs is merely *partial*, suggesting correctly that not all opvecs are comparable, comparisons between all *relevant* operations are produced: if two operations produce competing candidates, the partial order will determine which operation, and hence which candidate, wins.

This discovery of HS, that languages impose a partial order on vector-decomposable operations, yields a new way of viewing OT and addressing ongoing issues. Gen_{OT} can be reconstrued as consisting of the same operations as in HS, but multiply applied in any licit order. One can then assign an OT-opvec to each candidate, defined as the sum of the HS opvecs used in the candidate's construction. One can then ask and answer classic questions about the generative capacity of OT. For example, under what conditions does a theory of OT permit circular chain shifts? Precisely when two candidates constructed from corresponding sets of invertible operations have OT-opvecs that sum to zero.

Finally, the theory of opvecs and their orderings suggests an interesting line of further research. The opvec partial order may impose a derivational order on which operations are used in the earlier stages of a derivation and which are used in the latter stages. If this is the case, that there is a derivational order of operations, and this order and the operational vector order are related or the same or even similar, Harmonic Serialism may recapitulate many rule-based results in interesting ways, providing new insight into old problems.

References:

Pater, Joe. 2010. Serial Harmonic Grammar and Berber syllabification. In T. Borowsky, S. Kawahara, T. Shinya & M. Sugahara (eds.) Prosody Matters: Essays in Honor of Lisa Selkirk. Equinox Publishing.

Prince, Alan and Smolensky, Paul. 1993/2004. Optimality Theory : Constraint Interaction in Generative Grammar. Malden, MA: Blackwell Pub.

McCarthy, John J. 2000. Harmonic serialism and parallelism. In M. Hirotani, A. Coetzee, N. Hall, and J. Kim (eds.), Proceedings of the North East Linguistics Society 30. Amherst, MA: GLSA. Pp. 501-524

McCarthy, John. 2008. The serial interaction of stress and syncope. Natural Language & Linguistic Theory, 26:3, pp. 499-546.