A principled derivation of OT and HG within constraint-based phonology

Giorgio Magri (CNRS, SFL, University of Paris 8)

[1] We consider a set *Gen* of phonological mappings; a set $\mathbf{C}$ of n relevant constraints; and an order $\prec$ among arbitrary n-dimensional vectors ($\boldsymbol{x} \prec \boldsymbol{y}$ means vector $\boldsymbol{x}$ is smaller than vector $\boldsymbol{y}$). The **constraint-based grammar** $G_\prec$ sensibly realizes an underlying form x as a surface candidate $\mathsf{y} \in Gen(\mathsf{x})$ that is optimal because no other candidate $\mathsf{z} \in Gen(\mathsf{x})$ violates the constraints less. That is, no other candidate z has an n-dimensional vector of constraint violations that is smaller than the vector of constraint violations of the optimal candidate y when compared wrt order $\prec$.

[2] Most constraint-based phonological literature since the 80s has focused on the question of what are the right candidate and constraint sets *Gen* and $\mathbf{C}$. The question of what is the right order $\prec$ has instead been neglected: the debate has been limited to the comparison between OT's **lexicographic** order versus HG's **linear** order. Yet, the question concerning the order is just as fundamental. There are scores of other well defined orders $\prec$ among n-dimensional vectors. Why are we ignoring them and focusing only on OT's and HG's orders? This talk offers an answer.

[3] Let me introduce the intuition with three examples. • Suppose a grammar G realizes both $\mathbf{x}' =$ /ad/ and $\mathbf{x}'' =$ /tada/ faithfully as $\mathbf{y}' =$ [ad] and $\mathbf{y}'' =$ [tada]. Can I conclude that G realizes the underlying concatenation $\mathbf{x}' \cdot \mathbf{x}'' =$ /adtada/ faithfully as the surface concatenation $\mathbf{y}' \cdot \mathbf{y}'' =$ [adtada]? No: because G might ban clusters of obstruents that disagree in voicing and one such marked structure dt has been **created** by the concatenation of ad and tada into adtada. • Suppose a grammar G neutralizes $\mathbf{x}' =$ /adt/ to $\mathbf{y}' =$ [ad] and, say, faithfully realizes $\mathbf{x}'' =$ /ada/ as $\mathbf{y}'' =$ [ada]. Can I conclude that G realizes the underlying concatenation $\mathbf{x}' \cdot \mathbf{x}'' =$ /adtada/ non-faithfully as the surface concatenation $\mathbf{y}' \cdot \mathbf{y}'' =$ [adada]? No: because G might ban complex codas and one such marked structure has been **dissolved** (through proper syllabification) by the concatenation of adt and ada into adtada. • Suppose a grammar G realizes both $\mathbf{x}' =$ /adta/ and $\mathbf{x}'' =$ /da/ faithfully as $\mathbf{y}' =$ [adta] and $\mathbf{y}'' =$ [da]. Can I conclude that G realizes the underlying concatenation $\mathbf{x}' \cdot \mathbf{x}'' =$ /adtada/ as the surface concatenation $\mathbf{y}' \cdot \mathbf{y}'' =$ [adtada]? I submit yes, because the concatenation of adta and da into adtada is plausibly **innocuous**: it does not create nor dissolve any relevant marked structures.

[4] Let me formalize this intuition. • The concatenation $\mathbf{y}' \cdot \mathbf{y}''$ of two surface strings $\mathbf{y}'$ and $\mathbf{y}''$ is **innocuous** (wrt a markedness constraint set $\mathbf{M}$) provided it neither creates nor dissolves markedness violations. That is, neither $M(\mathbf{y}' \cdot \mathbf{y}'') > M(\mathbf{y}') + M(\mathbf{y}'')$ nor $M(\mathbf{y}' \cdot \mathbf{y}'') < M(\mathbf{y}') + M(\mathbf{y}'')$, whereby $M(\mathbf{y}' \cdot \mathbf{y}'') = M(\mathbf{y}') + M(\mathbf{y}'')$ for every markedness constraint M in $\mathbf{M}$. • The concatenation $\mathbf{x}' \cdot \mathbf{x}''$ of two underlying strings $\mathbf{x}'$ and $\mathbf{x}''$ is **innocuous** (wrt *Gen* and $\mathbf{M}$) provided the concatenation $\mathbf{y}' \cdot \mathbf{y}''$ of any two of their candidates $\mathbf{y}'$ and $\mathbf{y}''$ from $Gen(\mathbf{x}')$ and $Gen(\mathbf{x}'')$ is innocuous. To illustrate, the underlying strings /adta/ and /da/ with candidates obtained by changing obstruent voicing are innocuous relative to the constraints NoCompCoda, AgreeVoice, and NoVoice, as verified at the bottom of the page. • The intuition in **[3]** can now be formalized through the **axiom** that a grammar G be **concatenative** on innocuous concatenations: the surface realizations $G(\mathbf{x}' \cdot \mathbf{x}'')$ of the innocuous underlying concatenation $\mathbf{x}' \cdot \mathbf{x}''$ are the concatenations $G(\mathbf{x}') \cdot G(\mathbf{x}'')$ of the surface realizations $G(\mathbf{x}')$ and $G(\mathbf{x}'')$ of the two underlying strings in isolation, namely $G(\mathbf{x}' \cdot \mathbf{x}'') = G(\mathbf{x}') \cdot G(\mathbf{x}'')$. To illustrate, since the concatenation of /adta/ and /da/ is innocuous, the axiom requires that $G(\text{/adtada/}) = G(\text{/adta/}) \cdot G(\text{/da/})$.

[5] Some remarks are in order. • The concatenativity axiom is stated solely in terms of markedness constraints. In fact, most faithfulness constraints (Ident, Max, Dep, $\text{Max}_{[+\varphi]}$, $\text{Dep}_{[+\varphi]}$, Unif,

(/adtada/, [attata]) $\begin{bmatrix}0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}$ (/adtada/, [adtata]) $\begin{bmatrix}0\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}$ (/adtada/, [atdata]) $\begin{bmatrix}0\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}$ (/adtada/, [addata]) $\begin{bmatrix}0\\0\\2\end{bmatrix} = \begin{bmatrix}0\\0\\2\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}$ | (/da/, [ta]) $\begin{bmatrix}0\\0\\1\end{bmatrix}$

(/adtada/, [attada]) $\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}$ (/adtada/, [adtada]) $\begin{bmatrix}0\\1\\2\end{bmatrix} = \begin{bmatrix}0\\1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}$ (/adtada/, [atdada]) $\begin{bmatrix}0\\1\\2\end{bmatrix} = \begin{bmatrix}0\\1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}$ (/adtada/, [addada]) $\begin{bmatrix}0\\0\\3\end{bmatrix} = \begin{bmatrix}0\\0\\2\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}$ | (/da/, [da]) $\begin{bmatrix}0\\0\\0\end{bmatrix}$

(/adta/, [atta]) $\begin{bmatrix}0\\0\\0\end{bmatrix}$ (/adta/, [adta]) $\begin{bmatrix}0\\1\\1\end{bmatrix}$ (/adta/, [atda]) $\begin{bmatrix}0\\1\\1\end{bmatrix}$ (/adta/, [adda]) $\begin{bmatrix}0\\0\\2\end{bmatrix}$ NoCompCoda / AgreeVoice / NoVoice

INTE, etcetera) satisfy the identity $F(\mathbf{x}' \cdot \mathbf{x}'', \mathbf{y}' \cdot \mathbf{y}'') = F(\mathbf{x}', \mathbf{y}') + F(\mathbf{x}'', \mathbf{y}'')$ for just any underlying strings $\mathbf{x}', \mathbf{x}''$ and surface candidates $\mathbf{y}', \mathbf{y}''$ (provided the correspondence relation on the left-hand side is the union of the correspondence relations on the righthand side). • SPE grammars comply with the axiom by design, as the rule $A \to B/X\_Y$ only applies when the markedness constraint $*XAB$ is violated. • Constraint-based grammars as defined in **[1]** instead can flout the axiom. As a counterexample, let $\boldsymbol{x} \prec \boldsymbol{y}$ iff and only iff $\sum_{k=1}^{n} \frac{1}{k} x_k^2 < \sum_{k=1}^{n} \frac{1}{k} y_k^2$. We consider the three markedness constraints above plus IDENT. The constraint violation vectors are ordered as at the bottom of the page. We see that the constraint-based grammar $G_\prec$ realizes /da/ as [ta] and /adta/ as [atta] but realizes their concatenation /adtada/ as [attada] instead of [atta] · [ta].

[6] The main result of this paper is the following complete characterization of the orders $\prec$ that yield constraint-based grammars $G_\prec$ as in **[1]** that comply with the concatenativity axiom in **[4]**.

*An order $\prec$ among arbitrary n-dimensional vectors yields a constraint-based grammar $G_\prec$ that satisfies the concatenativity axiom if and only if there exist a certain number d (between 1 and n) of **weight** vectors $\boldsymbol{w}^{(1)} = (w_1^{(1)}, \ldots, w_n^{(1)}) \ldots \boldsymbol{w}^{(d)} = (w_1^{(d)}, \ldots, w_n^{(d)})$ such that two arbitrary n-dimensional vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$ satisfy the inequality $\boldsymbol{x} \prec \boldsymbol{y}$ if and only if there exists some index i (between 1 and d) such that:*

- *when we use the first $i-1$ weight vectors $\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(i-1)}$, the weighted sum of the components of $\boldsymbol{x}$ is equal to the weighted sum of the components of $\boldsymbol{y}$:*

$$\begin{aligned} \sum_{k=1}^{n} w_k^{(1)} x_k &= \sum_{k=1}^{n} w_k^{(1)} y_k \\ &\vdots \\ \sum_{k=1}^{n} w_k^{(i-1)} x_k &= \sum_{k=1}^{n} w_k^{(i-1)} y_k \end{aligned}$$

- *when we use instead the weight vector $\boldsymbol{w}^{(i)}$, the weighted sum of the components of $\boldsymbol{x}$ is strictly smaller than the weighted sum of the components of $\boldsymbol{y}$:*

$$\sum_{k=1}^{n} w_k^{(i)} x_k < \sum_{k=1}^{n} w_k^{(i)} y_k$$

[7] The complexity of this architecture is controlled by two parameters: the number d of weight vectors and the number s of non-zero components per weight vector. Thus, a simple but non-trivial implementation of this architecture is obtained by choosing the minimum value for one parameter (to achieve simplicity) and the maximum value for the other (to achieve non-triviality). We thus obtain two simplest non-trivial implementations.

[8] One simplest but non-trivial implementation of the boxed architecture corresponds to $d = 1$ and $s = n$: we use a unique weight vector $\boldsymbol{w}^{(d=1)} = \boldsymbol{w}$ but allow it to have the maximum number of non-zero components. In this case, the constraint-based grammar $G_\prec$ as in **[1]** corresponding to the order $\prec$ described in the box is the **HG grammar** corresponding to the weight vector $\boldsymbol{w}$.

[9] Next, we allow for maximum $d = n$ and thus use a full stack of n weight vectors $\boldsymbol{w}^{(d=1)}, \ldots, \boldsymbol{w}^{(d=n)}$. Yet, we require each of these n vectors to have only $s = 1$ component different from zero. We can assume without loss of generality that the unique non-zero components are all equal to one and that no two weight vectors $\boldsymbol{w}^{(i)}$ and $\boldsymbol{w}^{(j)}$ have the same component that is different from zero. Thus, the $d = n$ weight vectors induce the constraint ranking $C_{k_1} \gg C_{k_2} \gg \cdots \gg C_{k_n}$, where k_i is the index of the unique non-zero component of the weight vector $\boldsymbol{w}^{(i)}$. In this case, the constraint-based grammar $G_\prec$ as in **[1]** corresponding to the order $\prec$ described in the box is the **OT grammar** corresponding to this ranking $\gg$.

[10] In conclusion, HG and OT follow axiomatically as the simplest non-trivial implementations of the constraint-based architecture **[1]** which abide by the axiom on phonological behavior stated in **[4]**.

$$\underset{0.25}{\overset{[\text{atta}],[\text{ta}]}{\begin{bmatrix}0\\0\\0\\1\end{bmatrix}}} \prec \underset{0.333}{\overset{[\text{da}]}{\begin{bmatrix}0\\0\\1\\0\end{bmatrix}}} \prec \underset{0.583}{\overset{[\text{attada}]}{\begin{bmatrix}0\\0\\1\\1\end{bmatrix}}} \prec \underset{0.833}{\overset{[\text{adta}]}{\begin{bmatrix}0\\1\\1\\0\end{bmatrix}}} \prec \underset{1.000}{\overset{[\text{attata}]}{\begin{bmatrix}0\\0\\0\\2\end{bmatrix}}} \prec \underset{1.083}{\overset{[\text{adtata}]}{\begin{bmatrix}0\\1\\1\\1\end{bmatrix}}} \prec \underset{1.583}{\overset{[\text{adda}]}{\begin{bmatrix}0\\0\\2\\1\end{bmatrix}}} \prec \underset{1.833}{\overset{[\text{atda}]}{\begin{bmatrix}0\\1\\1\\2\end{bmatrix}}} \; \underset{1.833}{\overset{[\text{adtada}]}{\begin{bmatrix}0\\1\\2\\0\end{bmatrix}}} \prec \underset{2.333}{\overset{[\text{addata}]}{\begin{bmatrix}0\\0\\2\\2\end{bmatrix}}} \prec \underset{2.833}{\overset{[\text{atdada}]}{\begin{bmatrix}0\\1\\2\\2\end{bmatrix}}} \prec \underset{3.083}{\overset{[\text{atdata}]}{\begin{bmatrix}0\\1\\1\\3\end{bmatrix}}} \prec \underset{3.25}{\overset{[\text{addada}]}{\begin{bmatrix}0\\0\\3\\1\end{bmatrix}}} \begin{matrix}\text{COMPCODA}\\\text{AGREEVOICE}\\\text{NOVOICE}\\\text{IDENTVOICE}\end{matrix}$$