A principled derivation of Harmonic Grammar

Giorgio Magri
SFL, CNRS, University of Paris 8
magrigrg@gmail.com

Abstract

Phonologists focus on a few processes at the time. This practice is motivated by the intuition that phonological processes factorize into clusters with no interactions across clusters (e.g., obstruent voicing does not interact with vowel harmony). To formalize this intuition, we factorize a full-blown representation into under-specified representations, each encoding only the information needed by the corresponding phonological cluster. And we require a grammar for the original full-blown representations to factorize into grammars that handle the under-specified representations separately, independently of each other. Within a harmony-based implementation of constraint-based phonology, HG is shown to follow axiomatically from this grammar factorizability assumption.

1 Introduction

In constraint-based phonology, the best surface realization of an underlying form is the one with the smallest vector of constraint violations. How should constraint violation vectors be ordered to select the smallest? In other words, what is the proper model of constraint interaction? The literature has addressed this question by comparing competing ways of ordering constraint vectors on specific test cases. Yet, the predictions of a class of orderings on a specific test case depend on the choice of a specific constraint set. The conclusions reached are thus threatened to be overturned when a different constraint set is adopted.

An alternative, more principled approach starts instead from general formal properties that a grammar must satisfy in order to qualify as natural language phonology. And it deduces axiomatically from these desiderata what a suitable class of orderings of constraint violation vectors should look like. If this axiomatic deduction of the mode of constraint interaction holds independently of the constraint set, we will have untied the knot between the issue of determining the proper constraint set and the issue of characterizing the proper mode of constraint interaction.

This paper illustrates this research strategy. To set the background, section 2 recalls the framework of constraint-based phonology, independently of the choice of a specific mode of constraint interaction. Section 3 shows that a full-blown phonological representation often factorizes into multiple under-specified representations. And that these under-specified representations do not interact, in the sense that the constraint violations of the full-blown representations are simply the sum of the violations of the corresponding under-specified representations. In this case a grammar should factorize into multiple grammars that handle the under-specified representations separately, without these factor grammars interacting with each other. This factorizability condition formalizes the intuition that phonological processes factorize into small non-interacting clusters (e.g., obstruent voicing does not interact with vowel harmony), whereby phonologists can focus on a few processes at the time. Section 4 shows that a constraint-based grammar is indeed factorizable as long as the mode of constraint interaction satisfies a natural additivity condition. Section 5 finally shows that HG's weighted disharmony function can be derived axiomatically from this additivity condition. This result yields a principled justification of the HG mode of constraint interaction, which holds independently of any specific constraint set for any specific test case.

2 Constraint-based grammars

We assume that the core object of phonological theory is a **phonological mapping**, namely a pair (x, y) consisting of an underlying form x and a surface realization y (but see for instance Burzio 1996 for alternatives). To describe a specific phonological system, we start with a **representational framework** $\mathcal{R}$ that lists all the phonological mappings which are relevant for the system consid-

$$\mathcal{R} = \left\{ \begin{array}{cccc} (\text{/CV/, [CV]}) & (\text{/CV/, [CVC]}) & (\text{/CV/, [V]}) & (\text{/CV/, [VC]}) \\ (\text{/CVC/, [CV]}) & (\text{/CVC/, [CVC]}) & (\text{/CVC/, [V]}) & (\text{/CVC/, [VC]}) \\ (\text{/V/, [CV]}) & (\text{/V/, [CVC]}) & (\text{/V/, [V]}) & (\text{/V/, [VC]}) \\ (\text{/VC/, [CV]}) & (\text{/VC/, [CVC]}) & (\text{/VC/, [V]}) & (\text{/VC/, [VC]}) \end{array} \right\} \qquad \mathcal{C} = \left\{ \begin{array}{c} C_1 = \text{ONSET} \\ C_2 = \text{DEP} \\ C_3 = \text{CODA} \\ C_4 = \text{MAX} \end{array} \right\}$$

Figure 1: Representational framework $\mathcal{R}$ and constraint set $\mathcal{C}$ of the BSS (Prince and Smolensky 1993/2004).

ered. To illustrate, the representational framework $\mathcal{R}$ for the Basic Syllable System (BSS; Prince and Smolensky 1993/2004) consists of the sixteen mappings listed in figure 1.

We scan all the phonological mappings listed in a representational framework $\mathcal{R}$, extract their underlying forms, and collect them into the **base set** $B(\mathcal{R})$. To illustrate, the base of the BSS representational framework $\mathcal{R}$ in figure 1 consists of the four underlying syllable types /CV/, /CVC/, /V/, and /VC/. For every underlying form x in the base $B(\mathcal{R})$, we scan all the phonological mappings in $\mathcal{R}$ that feature this underlying form x and collect their surface forms into the **candidate set** $\mathcal{R}(\mathsf{x})$. To illustrate, the underlying forms of the BSS all share the candidate set consisting of the four surface syllable types [CV], [CVC], [V], and [VC].

A **constraint** C assigns to each phonological mapping (x, y) in the representational framework $\mathcal{R}$ a number $C(\mathsf{x}, \mathsf{y})$. This number $C(\mathsf{x}, \mathsf{y})$ is interpreted as a count of some undesirable phonological structure: an offending cluster, a mismatch between corresponding segments, etcetera. This number $C(\mathsf{x}, \mathsf{y})$ is thus assumed to be a nonnegative integer. This **constraint integrality assumption** captures the intuition that the properties relevant for phonology (contrary to phonetics) are discrete in nature. This assumption will play a crucial role in Section 5. We assume a set $\mathcal{C}$ consisting of a finite number n of constraints $C_1, \ldots, C_n$. It effectively represents a mapping (x, y) as the n-dimensional **constraint violation vector** $\mathbf{C}(\mathsf{x}, \mathsf{y}) = (C_1(\mathsf{x}, \mathsf{y}), \ldots, C_n(\mathsf{x}, \mathsf{y}))$. We denote by $\mathcal{C}(\mathcal{R})$ the set of the constraint vectors of all mappings in the representational framework $\mathcal{R}$. To illustrate, a constraint set for the BSS consists of the $n = 4$ constraints listed in figure 1.

The underlying and surface forms in the representational framework $\mathcal{R}$ are **discrete** objects (but see Smolensky, Goldrick, and Mathis 2014): finite strings constructed out of a finite number of discrete segments, or auto-segmental graphs constructed out of a finite number of feature values, etcetera. Dealing with discrete objects is difficult because only very little "structure" is defined on them. To circumvent this difficulty, constraint-based phonology "represents" the discrete phonological mappings in $\mathcal{R}$ as the set $\mathcal{C}(\mathcal{R})$ of numerical constraint violation vectors and thus imports into phonology the rich structure defined on numbers and vectors thereof (Haussler 1999).

For instance, numbers can be ordered based on their size. This ordering can be extended from single numbers to vectors in many different ways. Thus, let $\prec$ be some order defined among n-dimensional vectors. The inequality $\mathbf{a} \prec \mathbf{b}$ says that the vector $\mathbf{a}$ is **smaller** than the vector $\mathbf{b}$. The **constraint-based grammar** (CBG) corresponding to this order $\prec$ is the function $G_\prec = G_\prec^{\mathcal{R},\mathcal{C}}$ that realizes each underlying form x in the base $B(\mathcal{R})$ as the surface form $\mathsf{y} = G_\prec(\mathsf{x})$ in the candidate set $\mathcal{R}(\mathsf{x})$ with the smallest constraint violation vector $\mathbf{C}(\mathsf{x}, \mathsf{y})$. That is, the inequality $\mathbf{C}(\mathsf{x}, \mathsf{y}) \prec \mathbf{C}(\mathsf{x}, \mathsf{z})$ holds for every other candidate z in $\mathcal{R}(\mathsf{x})$ (we assume that such a candidate y always exists).

To illustrate, let us consider an arbitrary subset $S \subseteq \{1, ..., n\}$ that singles out the dimensions/constraints that are deemed relevant. The relation $\prec_S$ defined in (1) for any two vectors $\mathbf{a} = (a_1, ..., a_n)$ and $\mathbf{b} = (b_1, ..., b_n)$ is a partial order among n-dimensional vectors.

$$\mathbf{a} \prec_S \mathbf{b} \text{ iff } a_k \leq b_k \text{ for every } k \in S \tag{1}$$

We focus on the representational framework $\mathcal{R}$ and the constraint set $\mathcal{C}$ for the BSS in figure 1. We focus next on the order $\prec_S$ among 4-dimensional vectors corresponding to the set $S = \{C_1, C_3\}$. The corresponding CBG $G_{\prec_S}^{\mathcal{R},\mathcal{C}}$ maps all underlying forms to [CV]. This makes sense: if only the two markedness constraints C_1 = ONSET and C_3 = CODA are singled out as relevant by the set S, the smallest constraint vector is always the one corresponding to the unmarked surface form [CV].

3 Factorizable representations

3.1 Underspecification

A phonological representation x encodes a certain amount of phonological information. Often, this information can be split into two representations x'

$$\mathcal{R} = \left\{ \begin{array}{cccc} (/\mathsf{CV}/, [\mathsf{CV}]) & (/\mathsf{CV}/, [\mathsf{V}]) & (/\mathsf{V}/, [\mathsf{V}]) & (/\mathsf{V}/, [\mathsf{CV}]) \\ (/\mathsf{CV}/, [\mathsf{CVC}]) & (/\mathsf{CV}/, [\mathsf{VC}]) & (/\mathsf{V}/, [\mathsf{VC}]) & (/\mathsf{V}/, [\mathsf{CVC}]) \\ (/\mathsf{CVC}/, [\mathsf{CVC}]) & (/\mathsf{CVC}/, [\mathsf{VC}]) & (/\mathsf{VC}/, [\mathsf{VC}]) & (/\mathsf{VC}/, [\mathsf{CVC}]) \\ (/\mathsf{CVC}/, [\mathsf{CV}]) & (/\mathsf{CVC}/, [\mathsf{V}]) & (/\mathsf{VC}/, [\mathsf{V}]) & (/\mathsf{VC}/, [\mathsf{CV}]) \end{array} \right\} \begin{array}{c} \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \end{array} \left\{ \begin{array}{c} (/\square\mathsf{V}/, [\square\mathsf{V}]) \\ (/\square\mathsf{V}/, [\square\mathsf{VC}]) \\ (/\square\mathsf{VC}/, [\square\mathsf{VC}]) \\ (/\square\mathsf{VC}/, [\square\mathsf{V}]) \end{array} \right\} = \mathcal{R}''$$

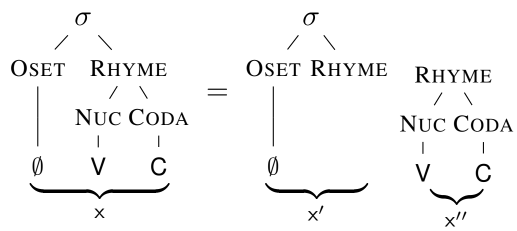
$$\begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \end{array}$$

$$\mathcal{R}' = \left\{ (/\mathsf{CV}\square/, [\mathsf{CV}\square]) \quad (/\mathsf{CV}\square/, [\mathsf{V}\square]) \quad (/\mathsf{V}\square/, [\mathsf{V}\square]) \quad (/\mathsf{V}\square/, [\mathsf{CV}\square]) \right\}$$

Figure 2: Factorization of the representational framework $\mathcal{R}$ of the BSS into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ under-specified for codas and for onsets, respectively

and x''. These two representations x' and x'' *individually* encode less information than the original representation x. In other words, they are **under-specified** relative to the original representation (Steriade 1995). Yet, these two under-specified representations x' and x'' *together* encode the same information as the full-blown representation x. In other words, the full-blown representation x **factorizes** into these two under-specified representations x' and x'', whereby we write $\mathsf{x} = \mathsf{x}'\mathsf{x}''$.

To illustrate again with the BSS, we note that a syllable type such as VC can be represented as the tree x on the left hand side of (2). This tree comes with the two sub-trees x' and x'' on the right hand side. These sub-trees can be interpreted as representations underspecified for codas and for onsets, respectively. We will denote these sub-trees compactly as V□ and □VC. The full-blown syllable x = VC thus factorizes into these two under-specified representations x' = V□ and x'' = □VC.



(2)

Feature-based phonology provides a natural strategy to factorize full-blown representations into under-specified representations. For instance, the round mid vowel can be represented as the tuple of feature values $\mathsf{x} = [+\mathsf{round} - \mathsf{high} - \mathsf{low}]$. This tuple comes with sub-tuples such as $\mathsf{x}' = [+\mathsf{round}]$ and $\mathsf{x}'' = [-\mathsf{high} - \mathsf{low}]$. These sub-tuples can be interpreted as representations under-specified for height and for rounding, respectively. The full-blown vowel x thus factorizes into these two under-specified representations x' and x''.

$$\underbrace{[+\mathsf{round}\ {-\mathsf{high}}\ {-\mathsf{low}}]}_{\mathsf{x}} = \underbrace{[+\mathsf{round}]}_{\mathsf{x}'}\underbrace{[-\mathsf{high}\ {-\mathsf{low}}]}_{\mathsf{x}''} \qquad (3)$$

3.2 Representational assumptions

A framework $\mathcal{R}$ of full-blown representations **factorizes** into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of under-specified representations provided $\mathcal{R}$ is the set of all and only the mappings $(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'')$ that factorize into two mappings $(\mathsf{x}', \mathsf{y}')$ and $(\mathsf{x}'', \mathsf{y}'')$ from $\mathcal{R}'$ and $\mathcal{R}''$, as in (4). In the sense that $\mathsf{x}'\mathsf{x}''$ and $\mathsf{y}'\mathsf{y}''$ are underlying and surface full-blown representations that factorize into the underlying and surface under-specified representations $\mathsf{x}', \mathsf{x}''$ and $\mathsf{y}', \mathsf{y}''$.

$$\mathcal{R} = \mathcal{R}'\mathcal{R}'' = \left\{ (\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'') \,\middle|\, \begin{array}{l} (\mathsf{x}', \mathsf{y}') \in \mathcal{R}' \\ (\mathsf{x}'', \mathsf{y}'') \in \mathcal{R}'' \end{array} \right\} \quad (4)$$

Equivalently, the base of the full-blown representational framework $\mathcal{R}$ factorizes into the bases of the under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$, namely $B(\mathcal{R}) = B(\mathcal{R}')B(\mathcal{R}'')$. And the candidate sets of $\mathcal{R}$ factorize into the corresponding candidate sets of $\mathcal{R}'$ and $\mathcal{R}''$, namely $\mathcal{R}(\mathsf{x}'\mathsf{x}'') = \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$.

To illustrate, let us consider again the representational framework $\mathcal{R}$ for the BSS in figure 1. We consider next the representational framework $\mathcal{R}'$ that consists of the four mappings that can be assembled out of the two representations CV□ and V□ that specify whether the onset is filled or empty but are under-specified for codas. And the representational framework $\mathcal{R}''$ that consists of the four mappings that can be assembled out of the two representations □V and □VC that specify whether the coda is filled or empty but are under-specified for onsets. As indicated by the dotted lines in figure 2, each full-blown mapping in $\mathcal{R}$ factorizes into the two under-specified mappings in $\mathcal{R}'$ and $\mathcal{R}''$ that sit in the same column and the same row. We conclude that condition (4) holds and that the framework $\mathcal{R}$ of full-blown syllable representations therefore factorizes into the two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of syllable representations under-specified for codas and for onsets.

As a second example, let us consider the representational framework $\mathcal{R}$ consisting of the 36

$$
\mathcal{R} = \left\{ \begin{array}{cccc}
(\text{/i/, [i]}) & (\text{/i/, [u]}) & (\text{/u/, [i]}) & (\text{/u/, [u]}) \\
(\text{/i/, [e]}) & (\text{/i/, [ɒ]}) & (\text{//u, [a]}) & (\text{/u/, [ɒ]}) \\
(\text{/i/, [a]}) & (\text{/i/, [ɒ]}) & (\text{/u/, [a]}) & (\text{/u/, [ɒ]}) \\
(\text{/e/, [i]}) & (\text{/e/, [u]}) & (\text{/o/, [i]}) & (\text{/o/, [u]}) \\
(\text{/e/, [e]}) & (\text{/e/, [o]}) & (\text{/o/, [e]}) & (\text{/o/, [o]}) \\
(\text{/e/, [a]}) & (\text{/e/, [ɒ]}) & (\text{/o/, [a]}) & (\text{/o/, [ɒ]}) \\
(\text{/a/, [i]}) & (\text{/a/, [u]}) & (\text{/ɒ/, [i]}) & (\text{/ɒ/, [u]}) \\
(\text{/a/, [e]}) & (\text{/a/, [o]}) & (\text{/ɒ/, [e]}) & (\text{/ɒ/, [o]}) \\
(\text{/a/, [a]}) & (\text{/a/, [ɒ]}) & (\text{/ɒ/, [a]}) & (\text{/ɒ/, [ɒ]}) \\
\vdots & \vdots & \vdots & \vdots
\end{array} \right\} \cdots \left\{ \begin{array}{c}
(\text{/+high, −low/, [+high, −low]}) \\
(\text{/+high, −low/, [−high, −low]}) \\
(\text{/+high, −low/, [−high, +low]}) \\
(\text{/−high, −low/, [+high, −low]}) \\
(\text{/−high, −low/, [−high, −low]}) \\
(\text{/−high, −low/, [−high, +low]}) \\
(\text{/−high, +low/, [+high, −low]}) \\
(\text{/−high, +low/, [−high, −low]}) \\
(\text{/−high, +low/, [−high, +low]})
\end{array} \right\} = \mathcal{R}''
$$

$$
\mathcal{R}' = \left\{ (\text{/−rnd/, [−rnd]}) \;\; (\text{/−rnd/, [+rnd]}) \;\; (\text{/+rnd/, [−rnd]}) \;\; (\text{/+rnd/, [+rnd]}) \right\}
$$

Figure 3: Factorization of the representational framework $\mathcal{R}$ for full-blown vowels into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ under-specified for height and for rounding, respectively

mappings that can be assembled out of the six vowels a, e, i, ɒ, o, and u. We consider next the representational framework $\mathcal{R}'$ that consists of the four mappings that can be assembled out of the two representations [+round] and [−round] underspecified for height. And the representational framework $\mathcal{R}''$ that consists of the nine mappings that can be assembled out of the the three representations [+high, −low], [−high, −low], and [−high, +low] underspecified for rounding. As indicated by the dotted lines in figure 3, each full-blown mapping in $\mathcal{R}$ factorizes into the two under-specified mappings in $\mathcal{R}'$ and $\mathcal{R}''$ that sit in the same column and the same row. We conclude that condition (4) holds and that the framework $\mathcal{R}$ of full-blown vowel representations therefore factorizes into the two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of vowel representations under-specified for height and for rounding.

3.3 No interaction

We consider a constraint set $\mathcal{C}$ for the mappings (x, y) in the full-blown representational framework $\mathcal{R}$. We assume that each constraint C in this constraint set $\mathcal{C}$ can be extended to the mappings $(\mathsf{x}', \mathsf{y}')$ and $(\mathsf{x}'', \mathsf{y}'')$ in the under-specified factor representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that condition (5) holds. It says that the number of violations $C(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y})$ that a constraint C assigns to a full-blown mapping $(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y})$ is the sum of the number of violations $C(\mathsf{x}', \mathsf{y}')$ and $C(\mathsf{x}'', \mathsf{y}'')$ that it assigns to the two under-specified factor mappings $(\mathsf{x}', \mathsf{y}')$ and $(\mathsf{x}'', \mathsf{y}'')$. In other words, no violations are created nor lost when the under-specified representations are assembled together into the full-blown representations.

$$C(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}) = C(\mathsf{x}', \mathsf{y}') + C(\mathsf{x}'', \mathsf{y}'') \qquad (5)$$

Suppose that this condition (5) holds for every underlying representation $\mathsf{x}'\mathsf{x}''$ in the base set $B(\mathcal{R}) = B(\mathcal{R}')B(\mathcal{R}'')$, for every surface representation $\mathsf{y}'\mathsf{y}''$ in the candidate set $\mathcal{R}(\mathsf{x}'\mathsf{x}'') = \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$, and for every constraint C in the constraint set $\mathcal{C}$. In other words, the set $\mathcal{C}(\mathcal{R})$ of constraint vectors of $\mathcal{R}$ is the sum of the sets $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$ of constraint vectors of $\mathcal{R}'$ and $\mathcal{R}''$, namely $\mathcal{C}(\mathcal{R}) = \mathcal{C}(\mathcal{R}') + (\mathcal{R}'')$. In this case, we say that the two under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ **do not interact** relative to the constraint set $\mathcal{C}$.

To illustrate, let us consider again the representational framework $\mathcal{R}$ and the constraint set $\mathcal{C}$ for the BSS in figure 1. The set $\mathcal{C}(\mathcal{R})$ of the constraint violation vectors of the sixteen mappings in the representational framework $\mathcal{R}$ is listed in figure 4. We extend the $n = 4$ constraints to the under-specified mappings in the two factor representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ straightforwardly as follows. The constraint $C_1 =$ ONSET assigns zero violations to the four mappings in the factor representational framework $\mathcal{R}''$ under-specified for onsets. The constraint $C_3 =$ CODA assigns zero violations to the four mappings in the other factor representational framework $\mathcal{R}'$ under-specified for codas. The other two constraints $C_2 =$ DEP and $C_4 =$ MAX simply count epenthetic and deleted consonants and thus assign violations to mappings in both factor representational frameworks. The corresponding sets $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$ of constraint vectors are listed at the bottom and on the left of figure 4. As indicated by the dotted lines, each constraint violation vector in $\mathcal{C}(\mathcal{R})$ is the (component-wise) sum of the constraint violation vectors in $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$ that sit in the

$$
\mathcal{C}(\mathcal{R}) = \left\{ \begin{array}{cccc}
\begin{array}{r} \textsc{Onset} \\ \textsc{Dep} \\ \textsc{Coda} \\ \textsc{Max} \end{array} \overset{\text{(/CV/,[CV])}}{\begin{bmatrix}0\\0\\0\\0\end{bmatrix}} & \overset{\text{(/CV/, [CVC])}}{\begin{bmatrix}0\\1\\1\\0\end{bmatrix}} & \overset{\text{(/CVC/, [CV])}}{\begin{bmatrix}0\\0\\0\\1\end{bmatrix}} & \overset{\text{(/CVC/, [CVC])}}{\begin{bmatrix}0\\0\\1\\0\end{bmatrix}} \\
\overset{\text{(/CV/, [V])}}{\begin{bmatrix}1\\0\\0\\1\end{bmatrix}} & \overset{\text{(/CV/, [VC])}}{\begin{bmatrix}1\\1\\1\\1\end{bmatrix}} & \overset{\text{(/CVC/, [V])}}{\begin{bmatrix}1\\0\\0\\2\end{bmatrix}} & \overset{\text{(/CVC/, [VC])}}{\begin{bmatrix}1\\0\\1\\1\end{bmatrix}} \\
\overset{\text{(/V/, [V])}}{\begin{bmatrix}1\\0\\0\\0\end{bmatrix}} & \overset{\text{(/V/, [VC])}}{\begin{bmatrix}1\\1\\1\\0\end{bmatrix}} & \overset{\text{(/VC/, [V])}}{\begin{bmatrix}1\\0\\0\\1\end{bmatrix}} & \overset{\text{(/VC/, [VC])}}{\begin{bmatrix}1\\0\\1\\0\end{bmatrix}} \\
\overset{\text{(/V/, [CV])}}{\begin{bmatrix}0\\1\\0\\0\end{bmatrix}} & \overset{\text{(/V/, [CVC])}}{\begin{bmatrix}0\\2\\1\\0\end{bmatrix}} & \overset{\text{(/VC/, [CV])}}{\begin{bmatrix}0\\1\\0\\1\end{bmatrix}} & \overset{\text{(/VC/, [CVC])}}{\begin{bmatrix}0\\1\\1\\0\end{bmatrix}}
\end{array} \right\}
\cdots\cdots
\left\{ \begin{array}{c}
\overset{\text{(/CV□/, [CV□])}}{\begin{bmatrix}0\\0\\0\\0\end{bmatrix}} \\
\overset{\text{(/CV□/, [V□])}}{\begin{bmatrix}1\\0\\0\\1\end{bmatrix}} \\
\overset{\text{(/V□/, [V□])}}{\begin{bmatrix}1\\0\\0\\0\end{bmatrix}} \\
\overset{\text{(/V□/, [CV□])}}{\begin{bmatrix}0\\1\\0\\0\end{bmatrix}}
\end{array} \right\} = \mathcal{C}(\mathcal{R}'')
$$

$$
\vdots \qquad \vdots \qquad \vdots \qquad \vdots
$$

$$
\mathcal{C}(\mathcal{R}') = \left\{ \overset{\text{(/□V/, [□V])}}{\begin{bmatrix}0\\0\\0\\0\end{bmatrix}} \quad \overset{\text{(/□V/, [□VC])}}{\begin{bmatrix}0\\1\\1\\0\end{bmatrix}} \quad \overset{\text{(/□VC/, [□V])}}{\begin{bmatrix}0\\0\\0\\1\end{bmatrix}} \quad \overset{\text{(/□VC/, [□VC])}}{\begin{bmatrix}0\\0\\1\\0\end{bmatrix}} \right\}
$$

Figure 4: The constraints for the representational framework $\mathcal{R}$ of the BSS can be extended to the factor frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that the constraint vectors in $\mathcal{C}(\mathcal{R})$ are the sums of the constraint vectors in $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$.

same column and the same row. We conclude that condition (5) holds and that the two under-specified frameworks $\mathcal{R}'$ and $\mathcal{R}''$ therefore do not interact relative to the constraint set $\mathcal{C}$.

4 Factorizable grammars

4.1 Factorizability

We consider a CBG $G_{\prec} = G_{\prec}^{\mathcal{R},\mathcal{C}}$ corresponding to some representational framework $\mathcal{R}$, some set $\mathcal{C}$ of n constraints for this representational framework $\mathcal{R}$, and some order $\prec$ among n-dimensional vectors. We assume that the full-blown representational framework $\mathcal{R}$ factorizes into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of under-specified representations and we consider some suitable extension of the constraint set $\mathcal{C}$ to $\mathcal{R}'$ and $\mathcal{R}''$. Using the same vector order $\prec$, we construct the CBGs $G'_{\prec} = G_{\prec}^{\mathcal{R}',\mathcal{C}}$ and $G''_{\prec} = G_{\prec}^{\mathcal{R}'',\mathcal{C}}$ for the under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$. We say that the original grammar $G_{\prec}$ **factorizes** into the two grammars $G'_{\prec}$ and $G''_{\prec}$ provided the identity (6) holds for any under-specified underlying representations x' and x'' in the base sets $B(\mathcal{R}')$ and $B(\mathcal{R}'')$ (see also Magri 2013). In this case, we also write $G_{\prec} = G'_{\prec}G''_{\prec}$.

$$G_{\prec}(\mathsf{x}'\mathsf{x}'') = G'_{\prec}(\mathsf{x}')G''_{\prec}(\mathsf{x}'') \tag{6}$$

This identity (6) says that an underlying representation $\mathsf{x}'\mathsf{x}''$ that factorizes into two under-specified underlying representations x' and x'' admits a surface realization $G_{\prec}(\mathsf{x}'\mathsf{x}'')$ that itself factorizes into the two under-specified surface representations $G'_{\prec}(\mathsf{x}')$ and $G''_{\prec}(\mathsf{x}'')$. In other words, the job done by the grammar $G_{\prec}$ can be outsourced to two grammars $G'_{\prec}$ and $G''_{\prec}$ that each carry out half of it independently from the other.

To illustrate, we consider again the representational framework $\mathcal{R}$ for the BSS and its factors $\mathcal{R}'$ and $\mathcal{R}''$ in figure 2. The grammar G in figure 5 tolerates empty onsets but deletes codas. We consider next the grammar G' for representations under-specified for codas that tolerates empty onsets. And the grammar G'' for representations under-specified for onsets that deletes codas. As indicated by the dotted lines, each full-blown mapping in G factorizes into the two under-specified mappings in G' and G'' that sit in the same column

$$G=\left\{\begin{array}{ll}(\text{/CV/, [CV]}) & (\text{/V/, [V]})\\ (\text{/CVC/, [CV]}) & (\text{/VC/, [V]})\end{array}\right\}\cdots\left\{\begin{array}{l}(\text{/□V/, [□V]})\\ \cdots(\text{/□VC/, [□V]})\end{array}\right\}=G''$$

$$\vdots \qquad \vdots$$

$$G'=\left\{(\text{/CV□/, [CV□]})\ (\text{/V□/, [V□]})\right\}$$

Figure 5: Factorization of the grammar G into two grammars G' and G''.

and the same row. We conclude that condition (6) holds and that the grammar G for full-blown syllable representations therefore factorizes into the two grammars G' and G'' for syllable representations under-specified for codas and for onsets.

Consider instead the grammar G in figure 6. It tolerates empty onsets and codas as long as they do not co-occur, as /VC/ is neutralized to [V] rather than faithfully realized as [VC]. This grammar does not factorize: onsets and codas cannot be handled independently. Indeed, it is easy to verify that, no matter what we replace the red question mark in figure 6 with, the factorizability identity (6) fails. This grammar G in figure 6 would be easy to get as a CBG corresponding to a markedness constraint set that contains a constraint that selectively penalizes the doubly-marked syllable type [VC]. But such a constraint does not satisfy the constraint condition (5): it would not penalize the underspecified surface representations $\mathsf{y}' =$ [V□] and $\mathsf{y}'' =$ [□VC] but it would penalize the corresponding full-blown representation $\mathsf{y}'\mathsf{y}'' =$ [VC]. In other words, the two underspecified representations do interact relative to such a constraint set.

4.2 Additive orders

An order $\prec$ among n-dimensional vectors is **additive** provided it satisfies the implication (7) for any three vectors **a**, **b**, and **c** (Anderson and Feil 1988). This implication (7) captures the intuition that, if **a** is smaller than **b** and if the same quantity **c** is added to both, the resulting sum $\mathbf{a}+\mathbf{c}$ ought to be smaller than the sum $\mathbf{b}+\mathbf{c}$ (all vector sums are component-wise).

$$\text{If } \mathbf{a} \prec \mathbf{b}, \text{ then } \mathbf{a}+\mathbf{c} \prec \mathbf{b}+\mathbf{c}. \qquad (7)$$

To illustrate, this additivity condition (7) is satisfied by the vector order $\prec_S$ defined in (1), for any choice of the set $S \subseteq \{1,\ldots,n\}$. Although this additivity condition (7) feels intuitive, it easy to construct orders that flout it. To illustrate, let $\mathbf{a} \prec \mathbf{b}$ provided the sum of squared components of the vector $\mathbf{a} = (a_1,\ldots,a_n)$ is smaller than the sum of squared components of the vector $\mathbf{b} = (b_1,\ldots,b_n)$, namely $a_1^2+\ldots+a_n^2 < b_1^2+\ldots+b_n^2$. The resulting order $\prec$ is not additive.

4.3 Establishing factorizability

The following proposition says that additivity of a vector order is sufficient to ensure that the corresponding CBG factorizes (see Prince 2015 for a special case of this result; see Magri and Storme 2020 for a different phonological justification of additive vector orders). Additivity can also be shown to be necessary, in the sense that for any order which is not additive we can construct a corresponding CBG that fails to factorize. Additivity thus provides a complete answer to the problem of characterizing grammatical factorizability.

Proposition 1 *Consider a framework $\mathcal{R}$ of full-blown representations that factorizes into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of under-specified representations, namely $\mathcal{R} = \mathcal{R}'\mathcal{R}''$ in the sense of condition (4) in subsection 3.2. Consider a set $\mathcal{C}$ of n constraints for the full-blown framework $\mathcal{R}$ that can be extended to the two under-specified frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that the additivity condition (5) in subsection 3.3 holds. Finally, consider an order $\prec$ among n-dimensional vectors that satisfies the additivity condition (7) in subsection 4.2. The corresponding CBG $G_{\prec}^{\mathcal{R},\mathcal{C}}$ for the full-blown representational framework $\mathcal{R}$ then factorizes into the two CBGs $G_{\prec}^{\mathcal{R}',\mathcal{C}}$ and $G_{\prec}^{\mathcal{R}'',\mathcal{C}}$ for the under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$.* □

To illustrate, we have seen in figure 2 that the representational framework $\mathcal{R}$ for the BSS factorizes into the two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of syllable representations under-specified for codas and for onsets, respectively. Furthermore, we have seen in figure 4 that the constraint set $\mathcal{C}$ for the BSS can be extended to these two under-specified frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that the additivity condition (5) holds. Finally, we have seen in subsection 4.2 that the vector order $\prec_S$ defined

$$G=\left\{\begin{array}{ll}(\text{/CV/, [CV]}) & (\text{/V/, [V]})\\ (\text{/CVC/, [CVC]}) & (\text{/VC/, [V]})\end{array}\right\}\cdots\left\{\begin{array}{l}(\text{/□V/, [□V]})\\ \cdots(\text{/□VC/, ??})\end{array}\right\}=G''$$

$$\vdots \qquad \vdots$$

$$G'=\left\{(\text{/CV□/, [CV□]})\ (\text{/V□/, [V□]})\right\}$$

Figure 6: An example of grammar G that does not factorize into two grammars G' and G''.

in (1) is additive for any subset S. Proposition 1 thus ensures that the CBG $G^{\mathcal{R},\mathcal{C}}_{\prec_S}$ factorizes.

4.4 Proof of proposition 1

Let us suppose that the two CBGs $G'_{\prec} = G^{\mathcal{R}',\mathcal{C}}_{\prec}$ and $G''_{\prec} = G^{\mathcal{R}'',\mathcal{C}}_{\prec}$ realize the under-specified underlying strings x' and x'' as the under-specified surface strings y' and y'' in the candidate sets $\mathcal{R}'(\mathsf{x}')$ and $\mathcal{R}''(\mathsf{x}'')$, namely $G'_{\prec}(\mathsf{x}') = \mathsf{y}'$ and $G''_{\prec}(\mathsf{x}'') = \mathsf{y}''$. The full-blown surface representation $\mathsf{y}'\mathsf{y}''$ belongs to the candidate set $\mathcal{R}(\mathsf{x}'\mathsf{x}'')$ because of the inclusion $\mathcal{R}(\mathsf{x}'\mathsf{x}'') \supseteq \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$. We need to show that $\mathsf{y}'\mathsf{y}''$ is indeed the surface realization of the full-blown underlying representation $\mathsf{x}'\mathsf{x}''$ according to the CBG $G_{\prec} = G^{\mathcal{R},\mathcal{C}}_{\prec}$, namely $G_{\prec}(\mathsf{x}'\mathsf{x}'') = \mathsf{y}'\mathsf{y}''$.

To this end, let us consider a candidate z in the candidate set $\mathcal{R}(\mathsf{x}'\mathsf{x}'')$ different from the candidate $\mathsf{y}'\mathsf{y}''$. This candidate z must factorize as $\mathsf{z} = \mathsf{z}'\mathsf{z}''$ into some candidates z' and z'' from $\mathcal{R}'(\mathsf{x}')$ and $\mathcal{R}''(\mathsf{x}'')$, because of the inclusion $\mathcal{R}(\mathsf{x}'\mathsf{x}'') \subseteq \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$. The assumption $\mathsf{z} \neq \mathsf{y}'\mathsf{y}''$ means that $\mathsf{z}' \neq \mathsf{y}'$ or $\mathsf{z}'' \neq \mathsf{y}''$ (or both). Without loss of generality, we assume $\mathsf{z}' \neq \mathsf{y}'$.

Since $\mathsf{z}' \neq \mathsf{y}'$, the assumption $G'_{\prec}(\mathsf{x}') = \mathsf{y}'$ says that the constraint violation vector $\mathbf{C}(\mathsf{x}', \mathsf{y}')$ of the winner mapping $(\mathsf{x}', \mathsf{y}')$ is smaller than the constraint violation vector $\mathbf{C}(\mathsf{x}', \mathsf{z}')$ of the loser mapping $(\mathsf{x}', \mathsf{z}')$, as in (8).

$$\mathbf{C}(\mathsf{x}', \mathsf{y}') \prec \mathbf{C}(\mathsf{x}', \mathsf{z}') \qquad (8)$$

Let us now turn to the other two candidates y'' and z''. If they are different as well, we reason analogously that the constraint violation vector $\mathbf{C}(\mathsf{x}'', \mathsf{y}'')$ of the winner mapping $(\mathsf{x}'', \mathsf{y}'')$ must be smaller than the constraint violation vector $\mathbf{C}(\mathsf{x}'', \mathsf{z}'')$ of the loser mapping $(\mathsf{x}'', \mathsf{z}'')$, as in (9).

$$\mathbf{C}(\mathsf{x}'', \mathsf{y}'') \prec \mathbf{C}(\mathsf{x}'', \mathsf{z}'') \qquad (9)$$

If instead these two candidates y'' and z'' are identical, their constraint violation vectors $\mathbf{C}(\mathsf{x}'', \mathsf{y}'')$ and $\mathbf{C}(\mathsf{x}'', \mathsf{z}'')$ coincide, as stated in (10).

$$\mathbf{C}(\mathsf{x}'', \mathsf{y}'') = \mathbf{C}(\mathsf{x}'', \mathsf{z}'') \qquad (10)$$

Since the order $\prec$ satisfies the additivity condition (7), the inequality (8) and the identity (10) can be summed together into the inequality (11).

$$\mathbf{C}(\mathsf{x}', \mathsf{y}') + \mathbf{C}(\mathsf{x}'', \mathsf{y}'') \prec \mathbf{C}(\mathsf{x}', \mathsf{z}') + \mathbf{C}(\mathsf{x}'', \mathsf{z}'') \qquad (11)$$

Suppose instead that it is the inequality (9) that holds rather than the identity (10). In this case, we note that the additivity condition (7) entails the variant in (12) for any four vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$. In fact, the assumption $\mathbf{a} \prec \mathbf{b}$ in the antecedent of (12) ensures that $\mathbf{a} + \mathbf{c} \prec \mathbf{b} + \mathbf{c}$ through the additivity condition (7). Analogously, the assumption $\mathbf{c} \prec \mathbf{d}$ ensures that $\mathbf{b} + \mathbf{c} \prec \mathbf{b} + \mathbf{d}$. The consequent $\mathbf{a} + \mathbf{c} \prec \mathbf{b} + \mathbf{d}$ then follows by transitivity of $\prec$.

$$\text{If } \mathbf{a} \prec \mathbf{b} \text{ and } \mathbf{c} \prec \mathbf{d}, \text{ then } \mathbf{a} + \mathbf{c} \prec \mathbf{b} + \mathbf{d} \qquad (12)$$

Since the vector order $\prec$ satisfies condition (12), the inequalities (8) and (9) can be summed together yielding once again the inequality (11).

By assumption, the two under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ do not interact relative to the constraint set $\mathcal{C}$, in the sense of condition (5). Thus, the sum of the constraint violation vectors $\mathbf{C}(\mathsf{x}', \mathsf{y}')$ and $\mathbf{C}(\mathsf{x}'', \mathsf{y}'')$ on the left hand side of the inequality (11) coincides with the constraint violation vector $\mathbf{C}(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'')$ of the corresponding full-blown mapping $(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'')$. Analogously for the right hand side, whereby the inequality (11) can be rewritten as (13).

$$\mathbf{C}(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'') \prec \mathbf{C}(\mathsf{x}'\mathsf{x}'', \mathsf{z}'\mathsf{z}'') \qquad (13)$$

By (13), the constraint violation vector of the candidate $\mathsf{y}'\mathsf{y}''$ is smaller than that of any competing candidate $\mathsf{z} = \mathsf{z}'\mathsf{z}''$. The CBG $G_{\prec}$ thus realizes the full-blown underlying representation $\mathsf{x}'\mathsf{x}''$ as the full-blown surface representation $\mathsf{y}'\mathsf{y}''$, namely $G_{\prec}(\mathsf{x}'\mathsf{x}'') = \mathsf{y}'\mathsf{y}''$ as desired.

5 HG and factorizability

5.1 Disharmony functions

Let us consider a particularly natural way of ordering numerical vectors. We start from a function H that assigns to each vector $\mathbf{a}$ a number $H(\mathbf{a})$ called its **disharmony**. Any two vectors $\mathbf{a}$ and $\mathbf{b}$ can then be ordered based on their disharmonies $H(\mathbf{a})$ and $H(\mathbf{b})$ as in (14): the smaller (and thus better) vector is the one with the smaller disharmony.

$$\mathbf{a} \prec_H \mathbf{b} \text{ iff } H(\mathbf{a}) < H(\mathbf{b}) \qquad (14)$$

The disharmony function H thus effectively defines a partial strict order $\prec_H$ among vectors.

Crucially, there exist numerical orders that are not induced by any disharmony function H. In the sense that condition (14) fails for some vectors, no matter how the disharmony function H is chosen. For instance, that can be shown to be case for the vector order $\prec_S$ defined in (1), whenever the set S has cardinality larger than one. The restriction to vector orders that are induced by disharmony functions is therefore substantive.

5.2 Additive disharmony functions

Proposition 1 says that the condition (7) that a vector order $\prec$ be additive is phonologically substantive because it ensures that the corresponding CBG $G_\prec$ factorizes. We are thus led to the following question: which assumptions on the disharmony function H suffice to ensure that the corresponding vector order $\prec_H$ defined through (14) satisfies this phonologically substantive additivity condition (7)? We will now see that it suffices to assume that the disharmony of the sum $\mathbf{a}+\mathbf{b}$ of two vectors $\mathbf{a}$ and $\mathbf{b}$ is equal to the sum of their disharmonies, as stated in (15).

$$H(\mathbf{a}+\mathbf{b}) = H(\mathbf{a}) + H(\mathbf{b}) \tag{15}$$

Indeed, let us assume that the numerical order $\prec_H$ induced by a disharmony function H satisfies the antecedent of the additivity implication (7), namely $\mathbf{a} \prec_H \mathbf{b}$. By definition (14), this means in turn that the disharmony $H(\mathbf{a})$ of the smaller vector $\mathbf{a}$ is smaller than the disharmony $H(\mathbf{b})$ of the larger vector $\mathbf{b}$, as in (16a). Let $H(\mathbf{c})$ be the disharmony of the vector $\mathbf{c}$. Whatever this number $H(\mathbf{c})$ looks like, it can be added to both sides of the disharmony inequality $H(\mathbf{a}) < H(\mathbf{b})$ without affecting it, yielding (16b). By the assumption (15) that the disharmony of a sum is the sum of the disharmonies, we can rewrite our inequality as in (16c). Finally, we use again the connection (14) between the disharmony function H and the corresponding numerical order $\prec_H$ to reinterpret the disharmony inequality $H(\mathbf{a}+\mathbf{c}) < H(\mathbf{b}+\mathbf{c})$ as the vector inequality $\mathbf{a}+\mathbf{c} \prec_H \mathbf{b}+\mathbf{c}$ required by the consequent of the additivity implication (7).

$$\begin{aligned} &\mathbf{a} \prec_H \mathbf{b} \iff \\ &\overset{(a)}{\iff} H(\mathbf{a}) < H(\mathbf{b}) \\ &\overset{(b)}{\iff} H(\mathbf{a}) + H(\mathbf{c}) < H(\mathbf{b}) + H(\mathbf{c}) \\ &\overset{(c)}{\iff} H(\mathbf{a}+\mathbf{c}) < H(\mathbf{b}+\mathbf{c}) \\ &\overset{(d)}{\iff} \mathbf{a}+\mathbf{c} \prec_H \mathbf{b}+\mathbf{c} \end{aligned} \tag{16}$$

5.3 Deriving HG's disharmony function

The two preceding subsections have motivated numerical orders defined though disharmony functions which satisfy the identity (15) whereby the disharmony of a sum of vectors is the sum of their disharmonies. We now explore the phonological implications of this assumption (15) by computing the disharmony of the constraint violation vector $\mathbf{C}(\mathsf{x},\mathsf{y})$ of an arbitrary mapping (x,y) as in (17).

In step (17a), we have recalled that the components of the constraint violation vector $\mathbf{C}(\mathsf{x},\mathsf{y})$ are the n constraint violations $C_1(\mathsf{x},\mathsf{y}),\dots,C_n(\mathsf{x},\mathsf{y})$. In step (17b), we have baroquely rewritten this constraint violation vector $\mathbf{C}(\mathsf{x},\mathsf{y})$ as the sum of many vectors: the vector with the 1st component equal to one and all other components equal to zero, repeated $C_1(\mathsf{x},\mathsf{y})$ times; the vector with the 2nd component equal to one and all other components equal to zero, repeated $C_2(\mathsf{x},\mathsf{y})$ times; and so on, down to the vector with the nth component equal to one and all other components equal to zero, repeated $C_n(\mathsf{x},\mathsf{y})$ times.

We now make the crucial assumption that the disharmony function H is additive. This means in particular that the disharmony of a sum of vectors is the sum of their disharmonies (the identity (15) extends trivially from two to an arbitrary finite number of vectors), yielding the identity (17c). Finally, let us call w_1 the disharmony of the vector with the 1st component equal to one and all other components equal to zero; let us called w_2 the disharmony of the vector with the 2nd component equal to one and all other components equal to zero; and so on. The disharmony of the constraint vector $\mathbf{C}(\mathsf{x},\mathsf{y})$ can thus be described as the sum of the constraint violations $C_1(\mathsf{x},\mathsf{y}),\dots,C_n(\mathsf{x},\mathsf{y})$ rescaled by $w_1,\dots,w_n$, as stated in (17d).

In conclusion, the reasoning in (17) shows that a disharmony function H that satisfies the additivity condition (15) is the one assumed in HG (Legendre et al. 1990b,a; Smolensky and Legendre 2006). And the HG **weights** $w_1,\dots,w_n$ can be interpreted as the disharmony of the **base vectors** that have one component equal to one and all other components equal to zero. These base vectors have no phonological meaning (they cannot be interpreted as constraint violation vectors). The reasoning in (17) thus illustrates the advantage of construing CBGs rather abstractly as in section 2, in terms of orders defined among arbitrary vectors.

5.4 The role of constraint integrality

As anticipated in section 2, the constraints used in phonology are assumed to only take (nonnegative) integer values, interpreted as numbers of violations. This assumption formalizes the intuition that the properties relevant to phonology are **discrete**—contrary to the properties relevant to phonetics, which are instead continuous and thus cannot be quantified through just integers. This **constraint integrality assumption** yields a number

$$
\begin{aligned}
H(\mathbf{C}(\mathsf{x},\mathsf{y})) &\overset{(a)}{=} H\left(\begin{bmatrix} C_1(\mathsf{x},\mathsf{y}) \\ C_2(\mathsf{x},\mathsf{y}) \\ \vdots \\ \vdots \\ C_n(\mathsf{x},\mathsf{y})\end{bmatrix}\right) \overset{(b)}{=} H\left(\underbrace{\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}+\ldots+\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}}_{C_1(\mathsf{x},\mathsf{y})\text{times}} + \underbrace{\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}+\ldots+\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}}_{C_2(\mathsf{x},\mathsf{y})\text{ times}} + \ldots + \underbrace{\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}+\ldots+\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}}_{C_n(\mathsf{x},\mathsf{y})\text{ times}}\right)\\
&\overset{(c)}{=} C_1(\mathsf{x},\mathsf{y})\, \underbrace{H\left(\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}\right)}_{w_1} + C_2(\mathsf{x},\mathsf{y})\, \underbrace{H\left(\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}\right)}_{w_2} + \ldots + C_3(\mathsf{x},\mathsf{y}) \underbrace{H\left(\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}\right)}_{w_n}\\
&\overset{(d)}{=} C_1(\mathsf{x},\mathsf{y})w_1 + C_2(\mathsf{x},\mathsf{y})w_2 + \ldots + C_n(\mathsf{x},\mathsf{y})w_n
\end{aligned} \tag{17}
$$

of finiteness effects when coupled with plausible assumptions on orders among n-dimensional vectors. For instance, Magri (2019) shows that (when coupled with a restriction to vector orders that are monotone), constraint integrality ensures that all candidate sets can be assumed to be finite without loss of generality.

The reasoning in (17) illustrates another finiteness effect of the constraint integrality assumption. Indeed, this reasoning crucially relies on the fact that the constraint violation vector $\mathbf{C}(\mathsf{x},\mathsf{y})$ can be expressed as a sum of a certain number of base vectors. Obviously, this decomposition is only possible because the components $C_k(\mathsf{x},\mathsf{y})$ of a constraint violation vector are integers but would fail otherwise.[1] The reasoning in (17) can thus be interpreted as another finiteness effect of the constraint integrality assumption: when this constraint integrality assumption is coupled with a restriction to numerical orders defined through additive disharmony functions, it ensures that the disharmony function admits a finite representation in terms of a finite number n of weights $w_1, \ldots, w_n$.

6 Conclusions

A phonological representation often factorizes into multiple under-specified representations that each encode only some of the information encoded by the original full-blown representation. We assume that these under-specified representations do not interact, in the sense that the number of constraint violations incurred by a mapping of full-blown representations coincides with the sum of the numbers of constraint violations incurred by the factor mappings of under-specified representations. In this case, we want a phonological grammar that handles full-blown representations to factorize into multiple grammars that handle the under-specified representations independently of each other. This paper has shown that **the HG implementation of constraint-based phonology follows from this factorizability desideratum plus the restriction to disharmony-based orders**. The latter assumption does not seem to admit a phonological justification but it is quite natural from a formal perspective. We conclude that HG admits a principled derivation from axioms that are phonologically or formally motivated (for alternative justifications of the HG framework, see Smolensky and Legendre 2006, and especially chapters 6 and 9). The proposed derivation crucially relies on the constraint-integrality assumption that phonologically relevant properties are discrete. Apart from this constraint-integrality assumption, the reasoning holds without any substantive assumptions on the constraint set.

[1] This reasoning (17) is essentially the proof of the Fundamental Theorem of Linear Algebra (Strang 2006), whereby a linear function between finite dimensional spaces admits a matrix representation. The only twist is that we do not need linearity (namely additivity plus homogeneity) but additivity suffices, because we are only dealing with integral vectors.

Acknowledgements

The research reported in this paper has been supported by a JCJC grant from the Agence Nationale de la Recherche (project title: *The mathematics of segmental phonotactics*).

References

Marlow Anderson and Todd Feil. 1988. *Lattice Ordered Groups: an Introduction*. D. Reidel Publishing Company.

Luigi Burzio. 1996. Surface constraints versus underlying representations. In Jacques Durand and Bernard Laks, editors, *Current trends in Phonology: Models and methods*, pages 97–122. University of Salford Publications, Paris and University of Salford.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California at Santa Cruz.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky. 1990a. Harmonic Grammar: A formal multi-level connectionist theory of linguistic wellformedness: An application. In *Proceedings of the 12th annual conference of the Cognitive Science Society*, pages 884–891, Hillsdale, NJ. Lawrence Erlbaum Associates.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky. 1990b. Harmonic Grammar: A formal multi-level connectionist theory of linguistic wellformedness: Theoretical foundations. In *Proceedings of the 12th annual conference of the Cognitive Science Society*, pages 388–395, Hillsdale, NJ. Lawrence Erlbaum.

Giorgio Magri. 2013. A note on the GLA's choice of the current loser from the perspective of factorizability. *Journal of Logic, Language, and Information*, 22:231–247.

Giorgio Magri. 2019. Finiteness of optima in constraint-based phonology. Manuscript, CNRS.

Giorgio Magri and Benjamin Storme. 2020. Constraint summation and the axiomatic derivation of HG. In *Proceedings of the 2019 Annual Meeting on Phonology*. Linguistic Society of America.

Alan Prince. 2015. One tableau suffices. Manuscript (Rutgers University). Available from the Rutgers Optimality Archive as ROA 1250.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in generative grammar*. Blackwell, Oxford. Original version, Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, April 1993. Available from the Rutgers Optimality Archive as ROA 537.

P. Smolensky, M. Goldrick, and D. Mathis. 2014. Optimization and quantization in gradient symbol systems: a framework for integrating the continuous and the discrete in cognition. *Cognitive Science*, 38.6:1102–1138.

Paul Smolensky and Géraldine Legendre. 2006. *The Harmonic Mind*. MIT Press, Cambridge, MA.

Donca Steriade. 1995. Underspecification and markedness. In John Goldsmith, editor, *The handbook of Phonological Theory*, pages 114–174. Blackwell, Oxford.

Gilbert Strang. 2006. *Linear Algebra and its applications*. Thomson Brooks/Cole.